Repertory of Digital Signal Processing Problems for Instructors

Author: Yiping Xie

Supervisor: Antoni Grau, Automatic Control Dept

Date of defense: 06/2017

Degree: Bachelor Degree in Informatics Engineering

Speciality: Computer Engineering

School: SCHOOL OF INFORMATICS OF BARCELONA (FIB)

University: UNIVERSITAT POLITECNICA DE CATALUNYA

(UPC)-Barcelona Tech

Abstract

Typically, DSP books are heavy, long and tedious to study for degree and master students. Moreover, it is hard for instructors to summarize the key parts of each chapter to give the right knowledge to students, sometimes giving them irrelevant parts, skipping the most interesting due to lack of time in lectures. To avoid these problems and in for the sake of a good learning process for master-degree students, this project tackles the gap between hard, long sections of exercises and the most profitable exercises to engage students in the field of DSP. To cover this objectives, the basic book of DSP is “Digital Signal Processing using MATLAB”, Ingle V.K., Proakis J.G. Brooks [1]. The idea is to create a collection of problems (statements/solutions) of all the chapters, from the most basic concept to more complex ideas of DSP. In the subject that instructors teach at the University, practical developments are done with specific hardware, particularly the C5515eZDSP USB Stick Development Tool [2]. It is also difficult for instructors to find suitable exercises to understand the classical DSP algorithms when real implementations are done. As a result, some hardware-oriented problems will also be included, which will be first solved in pseudo-code and then programmed on Code Composer Studio and verified on the DSP board.

Contents

1. STATE OF ART **1**

2. OBJECTIVES **1**

2.1 DIGITAL SIGNAL PROCESSING 1
2.2 MATLAB 2
2.3 CODE COMPOSER STUDIO 2
2.4 DSP DEVELOPMENT TOOL 2
2.5 OBJECTIVES 2
2.5.1 Repertory of problems/solutions 3
2.5.1.1 Design 3
2.5.1.2 Analysis 4
2.5.1.3 Implementation 4
2.5.1.4 Repertory 5
2.5.2 Hardware experiments 5
2.5.2.1 Design 5
2.5.2.2 Analysis 5
2.5.2.3 Implementation 6
2.5.2.3.1 DTMF Generation 6
2.5.2.3.2 DTMF Detection 7

3. WORK PLAN **8**

3.1 ESTIMATED TIME 9
3.2 GANTT CHART 9

4. METHODOLOGY **10**

5. PROJECT BUDGET **10**

5.1 CONSIDERATIONS 10
5.2 PROJECT BUDGET 10
5.2.1 Budget Monitoring 10
5.2.2 Human Resources Budget 10
5.2.3 Non-human Budget 11
5.2.3.1 Hardware Budget 11
5.2.3.2 Software Licenses Budget 11
5.2.4 Total Budget 12

6. ALTERNATIVE SOLUTIONS **12**

6.1. IMPSEQ AND STEPSEQ FUNCTIONS 12
6.2. CIRCONVF.M 13

7. RESULTS **15**

7.1 RESULTS OF REPERTORY 15
7.2 RESULTS OF DTMF GENERATION 15
7.3 RESULTS OF DTMF DETECTION 16

8. CONCLUSIONS **17**

9. ACKNOWLEDGEMENTS **18**

BIBLIOGRAPHY **19**

APPENDIX: THE REPERTORY **20**

CHAPTER 2 20

P2.1 .... *20*

Solutions 20

P2.2 .... *28*

Solutions 28

P2.3 .... *32*

Solutions 33

P2.4 .... *37*

Solutions 37

P2.5 .... *42*

Solutions 42

P2.6 .... *45*

Solutions 45

P2.7 .... *51*

Solutions 52

P2.8 .... *54*

Solutions 55

P2.9 .... *58*

Solutions 58

P2.10 .... *60*

Solutions 60

P2.11 .... *62*

Solutions 62

P2.12 .... *66*

Solutions 66

P2.13 .... *70*

Solutions 70

P2.14 .... *70*

Solutions 71

P2.15 .... *73*

Solutions 74

P2.16 .... *75*

Solutions 75

P2.17 .... *77*

Solutions 77

P2.18 .... *78*

Solutions 78

P2.19 .... *79*

Solutions 79

P2.20 .... *81*
Solutions 81
CHAPTER 3 85
P3.1 .... *85*
Solutions 85
P3.2 .... *92*
Solutions 92
P3.3 .... *94*
Solutions 94
P3.4 .... *101*
Solutions 101
P3.5 .... *107*
Solutions 108
P3.6 .... *109*
Solutions 109
P3.7 .... *110*
Solutions 110
P3.8 .... *112*
Solutions 112
P3.9 .... *115*
Solutions 115
P3.10 .... *119*
Solutions 119
P3.11 .... *128*
Solutions 128
P3.12 .... *134*
Solutions 135
P3.13 .... *135*
Solutions 135
P3.14 .... *142*
Solutions 142
P3.15 .... *144*
Solutions 145
P3.16 .... *147*
Solutions 148
P3.17 .... *148*
Solutions 148
P3.18 .... *154*
Solutions 154
P3.19 .... *161*
Solutions 161
P3.20 .... *165*
Solutions 166
P3.21 .... *166*

Solutions 166
P3.22 .... *171*
Solutions 172
CHAPTER 4 178
P4.1 .... *178*
Solutions 178
P4.2 .... *180*
Solutions 180
P4.3 .... *181*
Solutions 181
P4.4 .... *188*
Solutions 188
P4.5 .... *189*
Solutions 190
P4.6 .... *190*
Solutions 191
P4.7 .... *192*
Solutions 192
P4.8 .... *193*
Solutions 193
P4.9 .... *194*
Solutions 194
P4.10 .... *196*
Solutions 197
P4.11 .... *198*
Solutions 198
P4.12 .... *202*
Solutions 202
P4.13 .... *203*
Solutions 203
P4.14 .... *204*
Solutions 204
P4.15 .... *205*
Solutions 206
P4.16 .... *210*
Solutions 211
P4.17 .... *212*
Solutions 212
P4.18 .... *217*
Solutions 218
P4.19 .... *222*
Solutions 222
P4.20 .... *224*
Solutions 224

P4.21 .... *226*
Solutions 226
P4.22 .... *228*
Solutions 228
P4.23 .... *230*
Solutions 230
CHAPTER 5 232
P5.1 .... *232*
Solutions 233
P5.2 .... *234*
Solutions 234
P5.3 .... *235*
Solutions 236
P5.4 .... *239*
Solutions 239
P5.5 .... *241*
Solutions 242
P5.6 .... *244*
Solutions 244
P5.7 .... *246*
Solutions 247
P5.8 .... *248*
Solutions 248
P5.9 .... *250*
Solutions 250
P5.10 .... *251*
Solutions 252
P5.11 .... *257*
Solutions 257
P5.12 .... *259*
Solutions 259
P5.13 .... *260*
Solutions 260
P5.14 .... *262*
Solutions 262
P5.15 .... *263*
Solutions 264
P5.16 .... *265*
Solutions 265
P5.17 .... *269*
Solutions 269
P5.18 .... *273*
Solutions 274
P5.19 .... *275*

Solutions 276
P5.20 *278*
Solutions 279
P5.21 *281*
Solutions 281
P5.22 *282*
Solutions 282
P5.23 *283*
Solutions 283
P5.24 *284*
Solutions 284
P5.25 *284*
Solutions 285
P5.26 *286*
Solutions 286
P5.27 *287*
Solutions 287
P5.28 *288*
Solutions 288
P5.29 *293*
Solutions 294
P5.30 *295*
Solutions 295
P5.31 *295*
Solutions 295
P5.32 *301*
Solutions 302
P5.33 *304*
Solutions 304
P5.34 *305*
Solutions 306
P5.35 *311*
Solutions 312
P5.36 *314*
Solutions 315
P5.37 *320*
Solutions 321
P5.38 *323*
Solutions 323
P5.39 *325*
P5.40 *326*
P5.41 *326*
Solutions 326
CHAPTER 6 328

P6.1 .... *328*
Solutions 328
P6.2 .... *329*
Solutions 330
P6.3 .... *331*
Solutions 331
P6.4 .... *332*
Solutions 332
P6.5 .... *333*
Solutions 333
P6.6 .... *336*
Solutions 336
P6.7 .... *343*
Solutions 343
P6.8 .... *347*
Solutions 347
P6.9 .... *352*
Solutions 352
P6.10 .... *354*
Solutions 355
P6.11 .... *355*
Solutions 356
P6.12 .... *357*
Solutions 358
P6.13 .... *359*
Solutions 359
P6.14 .... *364*
Solutions 364
P6.15 .... *366*
Solutions 367
P6.16 .... *369*
Solutions 370
P6.17 .... *371*
Solutions 372
P6.18 .... *374*
Solutions 374
P6.19 .... *377*
Solutions 377
P6.20 .... *378*
Solutions 379
P6.21 .... *382*
Solutions 382
P6.22 .... *382*
Solutions 383

P6.23 .... *385*
Solutions 386
CHAPTER 7 387
P7.1 .... *387*
Solutions 387
P7.2 .... *388*
Solutions 388
P7.3 .... *389*
Solutions 390
P7.4 .... *391*
Solutions 391
P7.5 .... *393*
Solutions 393
P7.6 .... *395*
Solutions 395
P7.7 .... *398*
Solutions 398
P7.8 .... *401*
P7.9 .... *401*
Solutions 401
P7.10 .... *403*
Solutions 404
P7.11 .... *406*
Solutions 406
P7.12 .... *408*
Solutions 409
P7.13 .... *410*
Solutions 411
P7.14 .... *413*
Solutions 413
P7.15 .... *416*
Solutions 416
P7.16 .... *419*
Solutions 419
P7.17 .... *421*
Solutions 421
P7.18 .... *424*
Solutions 424
P7.19 .... *426*
Solutions 426
P7.20 .... *428*
Solutions 428
P7.21 .... *430*
Solutions 430

P7.22 ..... *433*
Solutions 433
P7.23 ..... *436*
P7.24 ..... *436*
Solutions 436
P7.25 ..... *439*
Solutions 439
P7.26 ..... *442*
Solutions 442
P7.27 ..... *444*
Solutions 445
P7.28 ..... *446*
Solutions 446
P7.29 ..... *447*
Solutions 448
P7.30 ..... *450*
Solutions 450
P7.31 ..... *454*
Solutions 454
P7.32 ..... *457*
Solutions 457
P7.33 ..... *459*
Solutions 459
P7.34 ..... *462*
Solutions 463
P7.35 ..... *465*
Solutions 465
P7.36 ..... *466*
Solutions 467
P7.37 ..... *476*
Solutions 476
P7.38 ..... *477*
Solutions 478
CHAPTER 8 481
EXAMPLE 8.1 ..... *481*
Solutions 481
EXAMPLE 8.2 ..... *481*
Solutions 481
EXAMPLE 8.3 ..... *482*
Solutions 482
EXAMPLE 8.4 ..... *482*
Solutions 482
EXAMPLE 8.5 ..... *483*
Solutions 484

EXAMPLE 8.6 ... *484*
Solutions ... 484
EXAMPLE 8.7 ... *486*
Solutions ... 486
EXAMPLE 8.8 ... *487*
Solution ... 487
EXAMPLE 8.9 ... *488*
Solutions ... 488
EXAMPLE 8.10 ... *489*
Solutions ... 489
EXAMPLE 8.11 ... *490*
Solutions ... 490
EXAMPLE 8.12 ... *492*
Solutions ... 492
EXAMPLE 8.13 ... *493*
Solutions ... 493
EXAMPLE 8.14 ... *494*
Solutions ... 495
EXAMPLE 8.15 ... *496*
Solutions ... 496
EXAMPLE 8.16 ... *496*
Solutions ... 496
EXAMPLE 8.17 ... *497*
Solutions ... 497
EXAMPLE 8.18 ... *498*
Solutions ... 498
EXAMPLE 8.19 ... *500*
Solutions ... 500
EXAMPLE 8.20 ... *501*
Solutions ... 501
EXAMPLE 8.21 ... *503*
Solutions ... 503
EXAMPLE 8.22 ... *504*
Solutions ... 504
EXAMPLE 8.23 ... *505*
Solutions ... 505
EXAMPLE 8.24 ... *506*
Solutions ... 506
EXAMPLE 8.25 ... *507*
Solutions ... 508
EXAMPLE 8.26 ... *509*
Solutions ... 509
EXAMPLE 8.27 ... *510*
Solutions ... 510

EXAMPLE 8.28 .... *511*
Solutions 511
EXAMPLE 8.29 .... *512*
Solutions 512
EXAMPLE 8.30 .... *514*
Solutions 514

1. State of Art

In this section I am going to briefly introduce some hints about the state of art of the solutions to post-class exercises of the textbooks, especially related to programming. As the teaching methods in signal processing have changed over the years from the simple “lecture-only” format to a more integrated “lecture-laboratory” environment, several textbooks in DSP have appeared that generally provide exercises that can be done using MATLAB, a software established as the standard for numerical computation in the signal-processing community and as a platform of choice for algorithm development [1]. However, few solutions to the exercises can be referred for students and practicing engineers interested in DSP. Nevertheless, I believe in a few years more collections of solutions/problems will be published to give a complete reference for learners interested in DSP.

2. Objectives

The objectives are mainly divided into two aspects: one part is collecting a report with solutions to the exercises in the DSP book including the MATLAB codes and figures of the results, the other part is designing hardware-oriented problems which will be programed on CCS (Code Composer Studio) and implemented on C5515eZSP board. First, I would like to give an introduction to the background and related terms.

2.1 Digital Signal Processing

Digital signal processing (DSP) is the use of digital processing, such as by computers, to perform a wide variety of signal processing operations. The signals processed in this manner are a sequence of numbers that represent samples of a continuous variable in a domain such as time, space, or frequency. Digital signal processing and analog signal processing are subfields of signal processing. DSP applications include audio and speech signal processing, sonar, radar and other sensor array processing, spectral estimation, statistical signal processing, digital image processing, signal processing for telecommunications, control of systems, biomedical engineering, seismic data processing, among others[3].

Digital signal processing can involve linear or nonlinear operations. Nonlinear signal processing is closely related to nonlinear system identification [4] and can be implemented in the time, frequency, and spatiotemporal domains. The application of digital computation to signal processing allows for many advantages over analog processing in many applications, such as error detection and correction in transmission as well as data compression [5]. DSP is applicable to both streaming data and static (stored) data [3].

2.2 MATLAB

The solutions for the problems in the textbook will be programed in MATLAB, a fourth-generation programming language, which allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, C#, Java, Fortran and Python[6].

2.3 Code Composer Studio

Code Composer Studio is an integrated development environment (IDE) that supports TI's Microcontroller and Embedded Processors portfolio. Code Composer Studio comprises a suite of tools used to develop and debug embedded applications. It includes an optimizing C/C++ compiler, source code editor, project build environment, debugger, profiler, and many other features. The intuitive IDE provides a single user interface taking you through each step of the application development flow. Familiar tools and interfaces allow users to get started faster than ever before. Code Composer Studio combines the advantages of the Eclipse software framework with advanced embedded debug capabilities from TI resulting in a compelling feature-rich development environment for embedded developers[7].

2.4 DSP Development Tool

The TMDX5515eZDSP is a small form factor, very low cost USB-powered DSP development tool which includes all the hardware and software needed to evaluate the industry's lowest power 16-bit DSP: TMS320C5515. This tool is similar to TMDX5505eZdsp in the form factor but provides more evaluation options such as USB2.0 and SD interface. The USB port provides enough power to operate the ultra-lowpower C5515 so no external power supply is required. This ultra low cost tool allows quick and easy evaluation of the advanced capabilities of the C5515, C5514, C5505A and C5504A processors. This tool has embedded XDS100 emulator for full source level debug capability and supports Code Composer Studio4.0™ integrated development environment (IDE) and eXpressDSP™ software which includes the DSP/BIOS™ kernel. The full contents of the Development Tool include: C5515 eZDSP board Code Composer Studio™ IDE Rev. 4.0 [8].

2.5 Objectives

The project consists on creating a collection of problems and solutions of all the chapters of the textbook, which can be divided into four phases, and hardware-oriented experiments implemented on DSP board, which can be divided into three phases.

2.5.1 Repertory of problems/solutions

2.5.1.1 Design

To cover this objectives, the textbook “Digital Signal Processing using MATLAB”, Ingle V.K./ Proakis J.G. Brooks[1], are organized in 8 chapters. The following is a list of chapters and a brief description of their contents.

Chapter 1, Introduction: This chapter introduces readers to the discipline of signal processing and presents several applications of digital signal processing, including musical sound processing, echo generation, echo removal, and digital reverberation. A brief introduction to MATLAB is also provided.

Chapter 2, Discrete-time Signals and Systems: This chapter provides a brief review of discrete-time signals and systems in the time domain. Appropriate use of MATLAB functions is demonstrated.

Chapter 3, The Discrete-time Fourier Analysis: This chapter discusses discrete-time signal and system representation in the frequency domain. Sampling and reconstruction of analog signals are also presented.

Chapter 4, The z-Transform: This chapter provides signal and system description in the complex frequency domain. MATLAB techniques are introduced to analyze z-transforms and to compute inverse z-transforms. Solutions of difference equations using the z-transform and MATLAB are provided.

Chapter 5, The Discrete Fourier Transform: This chapter is devoted to the computation of the Fourier transform and to its efficient implementation. The discrete Fourier series is used to introduce the discrete Fourier transform, and several of its properties are demonstrated using MATLAB. Topics such as fast convolution and fast Fourier transform are thoroughly discussed.

Chapter 6, Implementation of Discrete-Time Filters: This chapter discusses several structures for the implementation of digital filters. Several useful MATLAB functions are developed for the determination and implementation of these structures. Lattice and ladder filters are also introduced and discussed. In addition to considering various filter structures, we also treat quantization effects when finite-precision arithmetic is used in the implementation of IIR and FIR filters.

Chapter 7, FIR Filter Design: This chapter and the next introduce the important topic of digital filer design. Three important design techniques for FIR filters–namely, window design, frequency sampling design, and the equiripple filer design–are discussed. Several design examples are provided using MATLAB.

Chapter 8, IIR Filter Design: The chapter begins with the treatment of some basic filter types, namely, digital resonators, notch filters, comb filters, all-pass filters, and digital sinusoidal oscillators. This is followed by a brief description of the characteristics of three widely used analog filters. Transformations are described for converting these prototype analog filters into different frequency-selective digital filters. The chapter concludes with several IIR filter designs using MATLAB[1].

In every chapter, there are some examples which are provided with answers and solutions and post exercises which remain to be solved. All of them are helpful for the comprehension to the concepts of Digital signal Processing and conversance in using MATLAB to solve the related problems. Currently, there is no such a repertory of all the solutions of the post exercises in this book, so we decided to work on that in the period of four months. To have an intuition of the solutions, if possible, we divide the solutions into four parts: analytical calculation, MATLAB codes, MATLAB results and MATLAB figures.

2.5.1.2 Analysis

The main objective of this phase is to make an accurate analysis of the project and develop the consequent design. As mentioned in the previous document, the project can be divided into two aspects: the first part is a report collected with solutions to the problems in the DSP book including the MATLAB codes and the figures of results saved as vector graphics.

As for the final report, the project analysis will be necessary to define and set the objectives and requirements of the final report. Project design consists on solving problems through modeling in Digital Signal Processing area[9], which implies using all the knowledge acquired during the Bachelor Degree in Communications Engineering in order to use the different models to fit different problems.

2.5.1.3 Implementation

It is necessary to learn from the basic concepts to more complex ideas of Digital Signal Processing [3] and understand every problem that appears in the collection. What’s more, it is also helpful to learn all the examples with solutions given in every chapter to get familiar to the way of solving Digital Signal Processing problems using MATLAB, which means the combination of theory and practice. In every chapter, there is also some MATLAB functions written by the author of the textbook provided, which may be used frequently during solving the post exercises. As a result, to comprehend the MATLAB functions deeply and well is as well as significant to the next step. Only after ensuring to understand the concepts very well, I can be able to program on MATLAB to solve the problems and evaluate the merits of the methods.

Once I have figured out the various concepts of DSP the second phase will start. This stage will consist in numerical computation or model building and programming on MATLAB. I will work on the project chapter by chapter and if possible, the results will be shown in figures and saved as vector graphics, namely eps files for the final repertory[10][11][12].

2.5.1.4 Repertory

Finally, a repertory will be collected, including all the statements, solutions, namely the MATLAB codes, results shown in figures for the exercises in the textbook, which is also my final report of Bachelor Thesis.

2.5.2 Hardware experiments

2.5.2.1 Design

As for the experiments designed to solve hardware-oriented problems, the project analysis includes defining and setting objectives, requirements, features and the rationality of the experiments or systems. Furthermore, the state of art will be expanded and it is provided an analysis and evaluation of the different technologies and architectures used to the development. Project design consists on designing systems and creating the architectures implemented on DSP board, which requires extra knowledge about programming on Code Composer Studio, cross-compilation and debug between DSP board and upper computer.

The designing experiments should be involved in classic algorithms related to Digital Signal Processing and basic skills about Digital Signal Processor development. C5515eZDSP USB Stick is an industry's lowest power 16-bit processor helping conserve energy at exceptional levels and enabling longer battery life, which is especially suitable for audio signal processing applications, including voice recorder, musical instruments, portable medical solutions and so on[3].

To fulfill such requirements and to enlarge the features of C5515eZDSP USB Stick, I choose two applications as hardware-oriented experiments, which are Dual-Tone Multi-Frequency Signaling Generation and Detection by Goertzel Algorithm.

2.5.2.2 Analysis

The abbreviation DTMF stands for "*D*ual *T*one *M*ulti *F*requency", and is a method of representing digits with tone frequencies, in order to transmit them over an analog communications network, for example a telephone line. During development, care was taken to make use of all frequencies in the voice band, in order to reduce the demands placed on the transmission channel. In telephone networks, DTMF signals are used to encode dial trains and other information. Although the method used until now to form dial trains from a sequence of current pulses is still the standard in Germany, the transmission time is too long and places an unnecessary loading on the network. In addition, many telecommunications services are only available with the use of tone dialing[13].

DTMF (dual-tone multi-frequency signaling) Generation consists of the following objectives: to produce dual tones using sine wave generators, to generate the sequence of tones for a telephone number, to generate DTMF tones using C5515eZDSP USB Stick, to hear the results on headphones/computer loudspeakers.

DTMF Detection consists of the following objectives: to implement Goertzel Algorithm in C code, to use the Goertzel Algorithm to decode the tones generated by the DTMF Generator above, to display the number on the console of Code Composer Studio.

2.5.2.3 Implementation

2.5.2.3.1 DTMF Generation

For DTMF encoding, the digits 0-9 and the characters A-D, */E and #/F are represented as a combination of two frequencies[13]:

Frequency	1209 Hz	1336 Hz	1477 Hz	1633 Hz
697 Hz	1	2	3	A
770 Hz	4	5	6	B
852 Hz	7	8	9	C
941 Hz	*/E	0	#/F	D

With this system, the column is represented by a frequency from the upper frequency group (Hi-Group: 1209-1633 Hz), and the line by a frequency from the lower frequency group (Lo-Group: 697-941 Hz). The tone frequencies have been chosen such that harmonics are avoided. No frequency is the multiple of another, and in no case does the sum or difference of two frequencies result in another DTMF frequency.

As explained, DTMF signals are thus analog, and consist of two sine waves which are independent of each other. It is therefore not possible to generate them with only digital components. The digital signals must instead be converted by means of DACs (Digital-to Analog Converters)[14], which is integrated in the TLV320AIC3204 Stereo Coded on the C5515eZDSP USB Stick, and/or filters, into the desired sinusoidal waveforms.

The **ETSI ES 201 235[15]** has defined the requirements for a central office DTMF receiver to ensure reliable operations. For example, the receiver must tolerate slight variations (frequency bandwidths) in the eight frequencies and the relative signal amplitudes (twist) of the two frequencies comprising a valid digit. Also, the tone bursts must meet certain timing criteria such as on-off duration etc. The receiver must also reject speech signals and operate in the presence of certain noise levels, without incorrectly decoding the tone pairs[16]. Tone duration specified by the **ETSI ES 201 235** recommendation is at least 65 ms. Pause duration (a pause is a -80dBm level signal, encountered between two digit) must be of at least 65 ms.

In the implementation of DTMF Generation, I choose to set the tone duration as 70 ms and the pause duration 300ms. To generate the sine wave, the C code uses the **sine()** function in DSPLIB and the sampling frequency is 8000Hz. The tones are played for use as a signal for the DTMF Detection using Goertzel Algorithm.

2.5.2.3.2 DTMF Detection

For DTMF decoding, in order to receive DTMF signals with a Microcontroller without the use of costly special components, signals which are received need to be processed with an Analog-to-Digital Converter (ADC), which is integrated in the TLV320AIC3204 Stereo Coded on the C5515eZDSP USB Stick, and recognized with a digital filtering algorithm.

The minimum duration of a DTMF signal defined by the ETSI ES 201 235 standard is 65 ms. Therefore, there are at most 0.065 * 8000 = 520 samples available for estimation and detection. The DTMF decoder needs to estimate the frequencies contained in these short signals.

One common approach to this estimation problem is to compute the Discrete-Time Fourier Transform (DFT) samples close to the seven fundamental tones. For a DFT-based solution, it has been shown that using 205 samples in the frequency domain minimizes the error between the original frequencies and the points at which the DFT is estimated[17]. To minimize the error between the original frequencies and the points at which the DFT is estimated, I truncate the tones, keeping only 205 samples or 25.6 ms for further processing. At this point we could use the Fast Fourier Transform (FFT) algorithm to calculate the DFT. However, the popularity of the Goertzel algorithm in this context lies in the small number of points at which the DFT is estimated. In this case, the Goertzel algorithm is more efficient than the FFT algorithm[18]. The Goertzel algorithm is implemented in the form of a second-order IIR filter, the fomula for the second-order IIR filter is

$$H(z) = \frac{1}{1 - 2\cos\theta z^{-1} + z^{-2}}$$

$$\text{Where } \theta = 2\pi f / f_s$$

where f is the frequency of tone and fs is 8000Hz.
The difference equations for this IIR filter are

$$v(n) = 2\cos\theta v(n-1) - v(n-2) + x(n) \text{ where } n = 0, 1, 2..N$$

where x(n) is the filter input, v(n) is the filter output and N is the times filter is applied. However the outputs form IIR filter have both amplitude and phase shift, which means the filter outputs will have both real and imaginary components. To calculate the power in the signal we have power= $real^2$*imaginary^2, in this case we can ignore the phase shift and speed up the calculation. The power is calculated as follows[19]

$$|\,y(N)\,|^2 = v^2(N) + v^2(N-1) - 2\cos\theta v(N)v(N-1)$$

Where:

N = total number of times filter is used, typically 205
v(N) = filter output after N times
v(N-1) = previous filter output.

In the implementation, I design 8 IIR filters using MATLAB FDA Tool and generate the coefficients for CCS, one for each frequency and run each filter for the input signal a total of 205 times. The final time through, I calculate the power in each of the 8 frequencies and evaluate which pairs of tone are present from 8 calculated powers.
The calculated power should be high enough to be recognized as tones and the difference between the power of hi-frequency and lo-frequency should not be big enough. Ideally the two individual DTMF frequencies should have the same amplitude, however this is not always in the case. When transmitted along the telephone line, the amplitudes of DTMF may be different. The difference in amplitude of high and low frequencies is referred to "twist". As the ETSI ES 201 235 DTMF recommendations, the normal twist is less than 12db and the reverse twist is less than 6db[20].

3. Work Plan

As the initial methodology, the solutions of post exercises in the book of seven chapters, from Chapter 2 to Chapter 8, will be coded in MATLAB scripts, calling various functions in MATLAB library or written by the author or myself.

The estimated project duration is approximately 4 moths. The project starts on February 13th, 2017 and the deadline is on May 29th, 2017. It is important to consider that the initial planning could be revised and updated because of the evolution of the project.

As mentioned before, the agile methodology will allow us to revise and adapt dynamically the initial planning. Hence, if the stages referred in the previous point have different duration than expected, the planning will be modified. For example, if the stage has less duration than expected, it will start immediately the next one. Nonetheless, if the task lasts more than expected, it will delay the following tasks. At the end of each stage a meeting with the director of the project will take place in order to analyze the project and confirm that the author is following a good process.

Actually, due to the tight timetable, there is no sufficient time to finish the solutions for post exercises of Chapter 8, as a result, after the discussion between my tutor, Professor Antoni and me, we decide to collect statements and solutions of all the examples of Chapter 8 instead.

3.1 Estimated Time

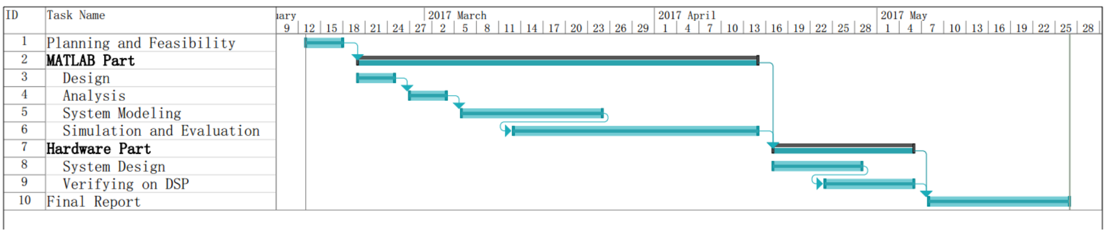
The four steps of the repertory part mentioned before need different duration time to be worked on. The Design part needs approximately one week, the Analysis part needs about another seven days, the Implementation part needs around sixty days and the Repertory part needs three weeks. The three steps of the hardware experiments part totally requires about three weeks be done with writing the repertory at the same time. As the following table, Table 1 is a chart to show the estimated time concretely.

Stage	Estimated dedication (hours)
Planning and Feasibility	40
Design	40
Analysis	40
System Modeling on MATLAB	160
Simulation and Evaluation on MATLAB	160
Design DTMF Generation and Detection System	60
Verifying on DSP Board	60
Final Report	120
Total	680 hours

Table 1: Estimated Time

The following is a Gantt Chart to show the project planning.

3.2 Gantt Chart



4. Methodology

In the analysis and implementation stages of repertory part, the solutions will be first calculated analytically and then coded in MATLAB scripts, calling various functions in MTALAB library or written by the author or myself. In the process of writing the final repertory the main tool is Microsoft Office and MathType to edit the fomulas.

In the implementation stage of hardware experiments, the model will be first simulated on MATLAB and then programmed in C Code on CCS. After the compilation without any error, the program will be downloaded to and run on the hardware, namely C5515eZDSA USB Stick, at the same time, the program will be debugged to ensure the correctness and accuracy as expected.

5. Project Budget

5.1 Considerations

In this section, an estimation of the cost of this project is presented taking into account respective human resources, hardware and software resources amortizations.

5.2 Project Budget

5.2.1 Budget Monitoring

In order to control de budget, at the end of each sprint the budget will be updated with the effective total amount of hours. Hence, the final budget with be a completely real budget based on real times. Specially, at the end of the second and eighth sprints the total amount of hours can widely vary.

5.2.2 Human Resources Budget

This project is going to be developed only by one person. Hence, this person will need to be both a project manager and a software programmer, as well as a hardware designer and developer. Thus, we will need to difference between each role in the total of 680 hours. In Table 2, an estimation of the cost is provided.

Role	Estimated	Estimated price per hour	Total estimated cost
Project Manager	40 hours	50 €/hour	2.000,00 €
Software Programmer	520 hours	35 €/hour	18.200,00 €
Hardware Designer and Developer	120 hours	35 €/hour	4.200,00 €
Total estimated	680 hours		24.400,00 €

Table 2: Human Resources Budget

5.2.3 Non-human Budget

5.2.3.1 Hardware Budget

In order to be able to design, implement and verify the hardware-oriented experiments, a set of hardware will be needed for different purposes. In Table 3, an estimation of the cost of that hardware is provided taking into account their useful life, as well as their amortizations. Considering the useful life of the hardware is five years, the final cost of the hardware will be as following.

Product	Price	Units	Useful life	Total estimated amortization
TMS320C5515eZdsp USB Stick Development Tool	280,00 €	1	5 years	28,00 €
ThinkPad T430s	500,00 €	1	5 years	50,00 €
Hann star LCD Monitor	114,00 €	1	5 years	11,40 €
Total estimated	894,00 €			89,40 €

Table 3: Hardware Budget

5.2.3.2 Software Licenses Budget

Additionally, some software products will be needed to carry out the project. Although some of them are available for free as this is an academic project, the real cost is considered. As in the hardware budget, their amortizations have been taken into account. In Table 4 the software budget is shown.

Product	Price	Units	Useful life	Total estimated amortization
Matlab R2014a	2.000,00 €	1	5 years	200,00 €
Code Composer Studio 6.1.1	300,00 €	1	4 years	37,50 €
Microsoft Office 2013 for Student	749,00 €	1	4 years	93,63 €
Windows 7 Utimate	69,99 €	1	5 years	Included on ThinkPad
Total estimated	3.118,99 €			331,13 €

Table 4: Software Licenses Budget

5.2.4 Total Budget

By adding all the budgets provided above, we can get the total estimated budget for this project, as shown in Table 5.

Concept	Estimated cost
Hardware	89,40 €
Software	331,13 €
Human resources	24.400,00 €
Total estimated cost	24.820,53 €

Table 5: Total budge

6. Alternative Solutions

As mentioned above, there are lots of MATLAB functions including in the MATLAB library and written by the author or myself are called when solving the post exercises in the MATLAB scripts. When using the MATLAB functions written by the author or writing the functions by myself, it can never be too careful to make sure the correctness and versatility. In the following solutions there maybe plenty of times to call the “homemade” functions, if there were any tiny bugs in these functions without awareness, the consequence could be unpredictable and unacceptable.

Here are several examples to illustrate what I mentioned above:

6.1. impseq and stepseq functions

In this case **impseq.m** and **stepseq.m**, shown as follows, are two functions written by the author to generate unit sample sequence and unit step sequence. There are several ways to generate these two types of basic sequence, and the author chooses an elegant way[1] of implementing $\delta(n)$ and $u(n)$, the logical relation n==0 and n>=0. However, a small problem of the implementation using logical relation is that the format of the sequence is logical.

```
function [x,n] = impseq(n0,n1,n2)
% Generates x(n) = delta(n-n0); n1 <= n <= n2
% ----------------------------------------------
% [x,n] = impseq(n0,n1,n2)
%
n = [n1:n2];x = [(n-n0)==0 ];
```

```
function [x,n] = stepseq(n0,n1,n2)
% Generates x(n) = u(n-n0); n1 <= n <= n2
% ------------------------------------------
% [x,n] = stepseq(n0,n1,n2)
%
n = [n1:n2]; x = [(n-n0)>=0];
```

In the solutions of P2.14, to prove the identity property of convolution,

$$x(n) * \delta(n - n_0) = x(n - n_0)$$

the MATLAB script needs to call the **conv_m(x,nx,h,nh)** function, which calls the **conv(a,b,shape)** function in MATLAB library to calculate the convolution of $x(n)$ and $\delta(n-n_0)$. The **conv(a,b,shape)** function requires the inputs "a" and "b" should be double format, which causes error because the $\delta(n-n_0)$ is in format of logical instead of double. So it is reasonable to add a line in the **impseq.m** and **stepseq.m** function that
change the format of them like following.

```
function [x,n] = impseq(n0,n1,n2)
% Generates x(n) = delta(n-n0); n1 <= n <= n2
% ----------------------------------------------
% [x,n] = impseq(n0,n1,n2)
%
n = [n1:n2];x = [(n-n0)==0 ];
x = double(x);
% convert x to double format from logical format.
```

```
function [x,n] = stepseq(n0,n1,n2)
% Generates x(n) = u(n-n0); n1 <= n <= n2
% ------------------------------------------
% [x,n] = stepseq(n0,n1,n2)
%
n = [n1:n2]; x = [(n-n0)>=0];
x = double(x);
% convert x to double format from logical format.
```

It is a simple example, but is shows the importance of taking care of the "homemade" functions.

6.2. circonvf.m

In the post exercises P5.26, the requirement is using the frequency domain approach, devise a MATLAB function to implement the circular convolution operation between two sequences. The format of the sequence should be

```
function x3 = circonvf(x1,x2,N)
% Circular convolution in the frequency domain
% x3 = circonvf(x1,x2,N)
% x3 = convolution result of length N
% x1 = sequence of length <= N
% x2 = sequence of length <= N
% N = length of circular buffer
```

The method of the function is first padding zeros for sequences x_1 and x_2 until their lengths are both N, and then calculating the DFT of x_1 and x_2, and finally calculating the IDFT of their product.

```
% Method:y(n) = idft(dft(x1)*dft(x2))
```

Normally, x_1 and x_2 are real-valued sequences, furthermore, due to the way of MATLAB's numeral calculation, it is a rather safer way to use the real part of y(n), instead of y(n) itself as the result of the circular convolution of x_1 and x_2. However, in this exercise, there is no specific requirement saying that x_1 or x_2 can not be complex-valued sequence, a better way is to add a conditional statement "if-else" statement to see whether x_1 and x_2 are real-valued or complex-valued and then decide whether to pick the real part as result, shown as followed:

```
function y = circonvf(x1,x2,N)
%
%function y=circonvf(x1,x2,N)
%
% N-point circular convolution between x1 and x2: (freq
domain)
% -------------------------------------------------------
------
% y : output sequence containing the circular convolution
% x1 : input sequence of length N1 <= N
% x2 : input sequence of length N2 <= N
% N : size of circular buffer
%
% Method: y(n) = idft(dft(x1)*dft(x2))
% Check for length of x1
if length(x1) > N
error('N must be >= the length of x1')
end
% Check for length of x2
if length(x2) > N
error('N must be >= the length of x2')
```

```
end
x1=[x1 zeros(1,N-length(x1))];
x2=[x2 zeros(1,N-length(x2))];
X1=fft(x1); X2=fft(x2);
if any(imag(x1) ~=0) || any(imag(x2)~=0)
y=ifft(X1.*X2);
else
y=real(ifft(X1.*X2));
end
```

To sum up, these are the only two cases I found because of the MATLAB Error or obviously wrong results in other MATLAB scripts. Through carefully analyzing, I think the revised alternatives are more suitable to solve the problems. Maybe there are still a few small bugs in the "homemade" functions to be discovered.

7. Results

7.1 Results of Repertory

The results of repertory are shown in the final part of this report as appendix.

7.2 Results of DTMF Generation

The DTMF tone sequence is first generated in MATLAB script **DTMFGenerator.m**, which generates a DTMF tones for sequence "1 2 3 4 5 6 7 8 9 * 0 #", with the duration of tone 70ms and duration of pause 300ms, added by 6 dB Gaussian White Noise by the MATLAB function **awgn(tones,snr).** Then the sequence will be played repeatedly by the MATLAB function **soundsc(tones,fs),** where fs is the sample frequency, namely 8000Hz. The result can be shown as follows[21]:

```
1 2 3 4 5 6 7 8 9 * 0 #
fx >>
```

After MATLAB simulation, the DTMF tones are generated by programing in C Code on CCS by means of finite state machine (FSM). The FSM consists of five states: START, BUTTON, SPACE, DONE, WAIT. When the result read from the phonenumber is not null, the pointer points to the next state, BUTTON, which calls the **generate_DTMF** function to generate the corresponded signal by means of calling the **sine()** function in DSPLIB. And the sequence will be transfer from digital signals to analog signals through codec TLV320AIC3204 and played via stereo out on the C5515eZDSP USB Stick. The console window of CCS can be shown as followed.

```
Console ⌧ Problems
DTMFGenerator [Project Debug Session] Texas Instruments XDS100v2 USB Emulator_0/C55xx: CIO (15:21:07)

PLL frequency 100 MHz
REGISTER --- CONFIG VALUES
PLL_CNTRL1   8be8 --- 8be8
PLL_CNTRL2   080e --- 0806 Test Lock Mon will get set after PLL is up
PLL_CNTRL3   8000 --- 8000
PLL_CNTRL4   0000 --- 0000


Running Project DTMF Generator
Generates DTMF frequencies for telephone number --> to Headphones/Lineout

Sampling frequency 8000 Hz Gain =  0 dB
```

7.3 Results of DTMF Detection

A 3.5mm audio cable line is used to transmit the DTMF generated by MATLAB into the stereo in on C5515eZDSP USB Stick as input, then be transferred from analog signals to digital signals. The input samples will be first processed by the function **goertzel_filter** until N is the typical number 205. Then the power of those outputs of the **goertzel_filter** function will be calculated separately due to different frequency. As long as the calculation is bigger than the corresponded threshold, the corresponded bit of the **goertzel_bit_mask** will be set. An example is as follows[21].

```
if ( goertzel_output_697_Hz > GOERTZEL_THRESHOLD)
 {
  goertzel_bit_mask |= 0x0001; /* Set bit 0 in bit mask */
 }
else
 {
  goertzel_bit_mask &= 0xFFFE; /* Clear bit 0 in bit mask */
 }
```

After that, there will be a set of “if-else” statements to determine which button is pressed and at the same time whether the twist is followed the recommendations. An example is as followed:

```
if (( BUTTON_0 == goertzel_bit_mask)&&(goertzel_output_941_Hz>=goertzel_output_1336_Hz*0.398)&&(goertzel_output_1336_Hz>=goertzel_output_941_Hz*0.158))
  {
   puts("Button 0 pressed.\n");
  }
```

The results shown in the console window of CCS, which can see that the Goertzel Algorithm is able to detect the sequence added to 6 db Gaussian White Noise, without any errors.

```
GoertzelAlgorithm [Project Debug Session] Texas Instruments XDS100v2 USB Emulator_0/C55xx: CIO (15:25:20)

PLL frequency 100 MHz
REGISTER --- CONFIG VALUES
PLL_CNTRL1   8be8 --- 8be8
PLL_CNTRL2   080e --- 0806 Test Lock Mon will get set after PLL is up
PLL_CNTRL3   8000 --- 8000
PLL_CNTRL4   0000 --- 0000
Sampling frequency 8000 Hz Gain = 24 dB

Running Goertzel Algorithm Project

Input: Audio from telephone keyboard to microphone
Output: Goertzel filtered output on headphones
```

```
Console ☒ Problems
GoertzelAlgorithm [Project Debug Session] Texas Instruments XDS100v2 USB Emulator_0/C55xx: CIO (15:25:20)
Button 1 pressed.

Button 2 pressed.

Button 3 pressed.

Button 4 pressed.

Button 6 pressed.

Button 7 pressed.

Button 8 pressed.

Button 9 pressed.

Button * pressed.

Button 0 pressed.

Button # pressed.
```

8. Conclusions

In this project, two main parts of work are done according to the timetable in 4 months. A repertory of Digital Signal Processing based on “Digital Signal Processing using MATLAB”, Ingle V.K., Proakis J.G. Brooks[1] is written well-organized and the results as well as solutions are properly-evaluated and correctly-tested on MATLAB. A bunch of analytical results, MATLAB calculation results, MATLAB figures are collected with the post exercises together, to be a final report. What’s more, a set of hardware-oriented experiments are designed. The DTMF tones are well generated on MATLAB and also on CCS by downloading the programs to C5515eZDSP USB Stick. At the same time, DTMF Detection using Goertzel Algorithm is well implemented and tested which is able to recognize correctly form DTMF tones to show which the buttons are pressed. By finishing this report, I learned deeply from the basic concepts to increased level of complex ideas of DSP, achieved a very good level of MATLAB coding, acquired basic skills in DSP Development field. Moreover, The final report is able to help instructors summarize the key parts of each chapter to give the right knowledge to students and find suitable exercises to understand the classical DSP algorithm when real implementations are done[22]. This final report is also able to help students on DSP class or people who are interested in DSP to get access to the reference of solutions of post class exercises and tackle the gap between hard, long sections of exercises and the most profitable exercises, which will support students a good learning process in DSP field.

9. Acknowledgements

First I am deeply indebted to my family, including my father, my little brother and especially my mother, who supports me not only in finance but stands by me mentally all the time. I am also indebted to my tutor, Prof. Antoni Grau, who gave me a lot of patient guidance and thorough support in this four months, which is the key of finishing this project favorably. I am also indebted to all related school stuff in both Beihang University and FIB, UPC, without whose help, I wouldn’t make it to such a nice exchange program finishing my Bachelor Thesis. Among the school stuff, I am particularly indebted to Tong Jun, from International Division of Beihang University, who guided me with all the procedures of this exchange program, and Berenice Martin as well as Maria Eulalia Minarro, from International Office of FIB, UPC, who helped me a lot during the process of the re-application passport due to lost and getting the certificate with my NIE number on. I am also indebted to my roommates, Xunyu Lin, Ruochen Wen, Yunhong Wang, who took good care of me and had a great time seeing the beautiful scenery of Barcelona together on weekends. Particularly, I am indebted to Xunyu Lin, who gave me advice on learning in the field of Machine Learning, which is new for me but also is what I am interested in and going to learn in the master degree. I am also indebted to Edmundo Guerra, the Doctor student in my tutor’s lab, for giving me countless help and suggestions to make me get along with the new environment. Last but not the least, I am incredibly indebted to my girlfriend, Jiaying Yang, for listening to me and giving me sincere advice when I had troubles, no matter in the studying or in the daily life, and for the patience and company of me in the four months regardless the time difference or long distance. Finally I sincerely thank everyone that helped with this exchange program and Bachelor Program and wish them good luck in the future.

Bibliography

[1] Digital Signal Processing using MATLAB, Ingle V.K., Proakis J.G. Brooks, Cole Thmson Learning. 2000.
[2] http://www.ti.com/tool/tmdx5515ezdsp
[3] https://en.wikipedia.org/wiki/Digital_signal_processing
[4] Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains, Billings, Stephen A, UK: Wiley.2013.
[5] Digital Signal Processing: Instant access, Broesch James D, Stranneby Dag, Walker William. 2008.
[6] https://en.wikipedia.org/wiki/MATLAB
[7] https://en.wikipedia.org/wiki/Code_Composer_Studio
[8] TMS320C5515 eZdspTM USB Stick Technical Reference, 512845-0001, RevA. Spectrum Digital, Inc., 2010.
[9] Digital Signal Processing. Principles, Algorithms and Applications, John G. Proakis and Dimitris Manolakis, 4th Edition, Pearson Prentice Hall.2006.
[10] Computer-Based Exercises for Signal Processing using MATLAB ver. 5, McClellan, .H.; Burrus, C.S.; Oppenheim A.V., Parks T.W., Schafer R.W. Schuessler H.W., PrenticeHall. 1998.
[11] Digital Filters and Signal Processing with Matlab Exercises, Jackson L.B. Kluwer Academic Publishers. 3rd Edition. 1995.
[12] Digital Signal Processing: a computer-based approach, Sanjit K. Mitra, 3rd Edition, McGraw-Hill. 2005
[13] Generation and Recognition of DTMF Signals with the Microcontroller MSP430
[14] Design and Configuration Guide for the TLV320AIC3204 and TLV320AI3254 Audio Codecs, Texas Instrument. 2010
[15] ETSI ES 201 235-2 V1.1.1
[16] DTMF Tone Gneration. An Implementation using the TMS320C2xx, Texas Instrument.1997.
[17] Digital Signal Processing: a practical approach, Ifeachor, E.C; Jervis, B,W,. Addison-Wesley. 1997.
[18] Digital Signal Processing Fundamentals and Applications, Li Tan, DeVry University, Decatur, Georgia.2013.
[19] Modified Goertzel Algorithm in DTMF Detection Using the TMS320C80. Texas Instrument.1996.
[20] http://www.britishtelephones.com/dtmf.htm
[21] TMS320C55xx Optimizing C/C++ Compiler User's Guide, Texas Instrument. 2010.
[22] Understanding Digital Signal Processing, Richard G. Lyons, 3rd Ed, Prentice Hall. 2010.

Appendix: The Repertory

Chapter 2

P2.1

Generate the following sequences using the basic MATLAB signal functions and the basic MATLAB signal operations discussed in this chapter. Plot signal samples using the **stem** function.

1. $x_1(n) = 3\delta(n + 2) + 2\delta(n) - \delta(n - 3) + 5\delta(n - 7)$, $-5 \leqslant n \leqslant 15$.
2. $x_2(n) \sum_{k=-5}^{5} e^{-|k|}=\delta(n - 2k)$, $-10 \leqslant n \leqslant 10$.
3. $x_3(n) = 10u(n) - 5u(n - 5) - 10u(n - 10) + 5u(n - 15)$.
4. $x_4(n) = e^{0.1n}[u(n + 20) - u(n - 10)]$.
5. $x_5(n) = 5[\cos(0.49\pi n) + \cos(0.51\pi n)]$, $-200 \leqslant n \leqslant 200$. Comment on the waveform shape.
6. $x_6(n) = 2 \sin(0.01\pi n)\cos(0.5\pi n)$, $-200 \leqslant n \leqslant 200$. Comment on the waveform shape.
7. $x_7(n) = e^{-0.05n} \sin(0.1\pi n + \pi/3)$, $0 \leqslant n \leqslant 100$. Comment on the waveform shape.
8. $x_8(n) = e^{0.01n} \sin(0.1\pi n)$, $0 \leqslant n \leqslant 100$. Comment on the waveform shape.

Solutions

1. $x_1(n) = 3\delta(n + 2) + 2\delta(n) - \delta(n - 3) + 5\delta(n - 7)$, $-5 \leq n \leq 15$

```
% P2.1
%% P0201a: x1(n) = 3*delta(n+2) + 2*delta(n) - delta(n-3)
% + 5*delta(n-7), -5<= n <= 15;
clc; close all;
x1 = 3*impseq(-2,-5,15) + 2*impseq(0,-5,15) - impseq(3,-
5,15) ...
+ 5*impseq(7,-5,15);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0201a'); n1 = [-
5:15];
Hs = stem(n1,x1,'filled'); set(Hs,'markersize',2);
axis([min(n1)-1,max(n1)+1,min(x1)-1,max(x1)+1]);
xlabel('n','FontSize',8); ylabel('x_1(n)','FontSize',8);
title('Sequence x_1(n)','FontSize',8);
set(gca,'XTickMode','manual','XTick',n1,'FontSize',8);
print -deps2 ../EPSFILES/P0201a;
```

The plots of $x_1(n)$ is shown in Figure 2.1.

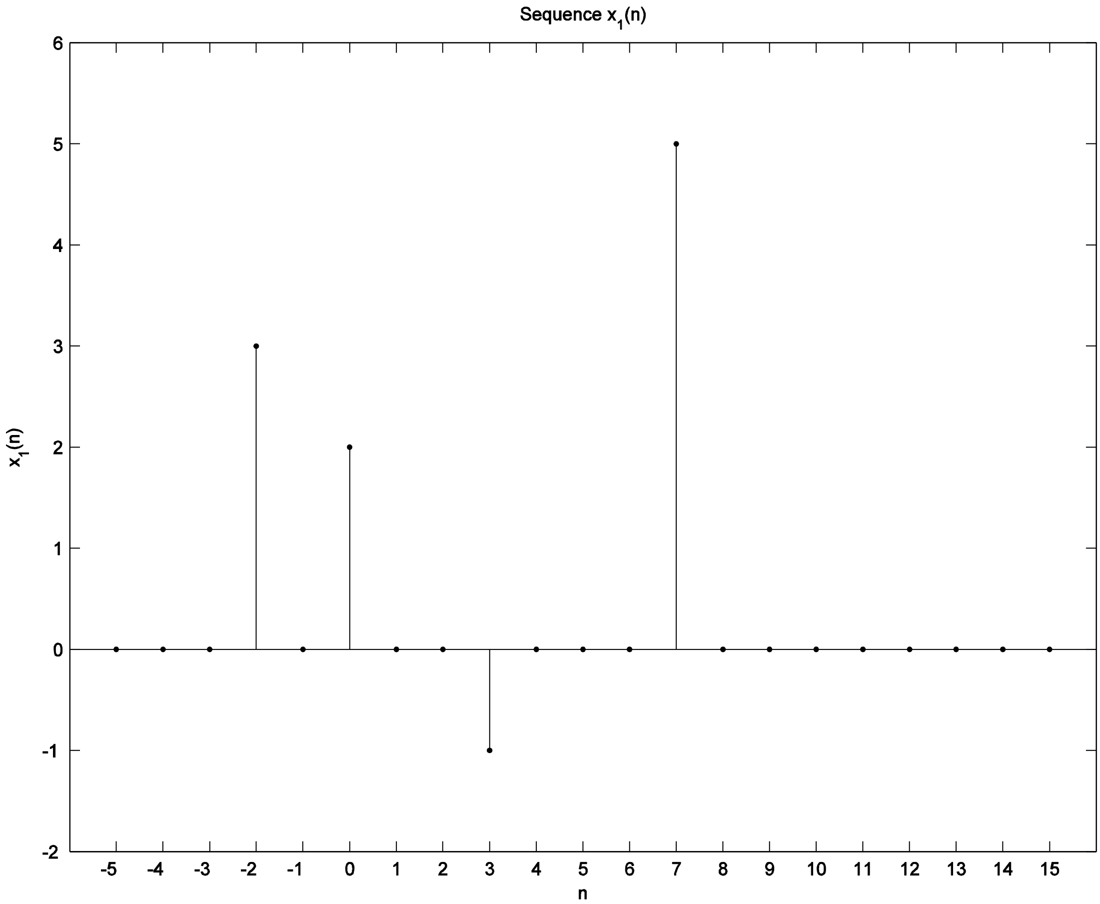


Figure 2.1: Problem P2.1.1 sequence plot

2. $x_2(n) = \sum_{k=-5}^{5} e^{-|k|}\delta(n-2k), -10 \le n \le 10.$

```
%% P0201b: x2(n) = sum_{k = -5}^{5} e^{-|k|}*delta(n -
% 2k), -10 <= n <= 10
clc; close all;
n2 = [-10:10]; x2 = zeros(1,length(n2));
for k = -5:5
x2 = x2 + exp(-abs(k))*impseq(2*k ,-10,10);
end
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0201b');
Hs = stem(n2,x2,'filled'); set(Hs,'markersize',2);
axis([min(n2)-1,max(n2)+1,min(x2)-1,max(x2)+1]);
xlabel('n','FontSize',8); ylabel('x_2(n)','FontSize',8);
title('Sequence x_2(n)','FontSize',8);
set(gca,'XTickMode','manual','XTick',n2);
print -deps2 ../EPSFILES/P0201b;
```

The plots of $x_2(n)$ is shown in Figure 2.2

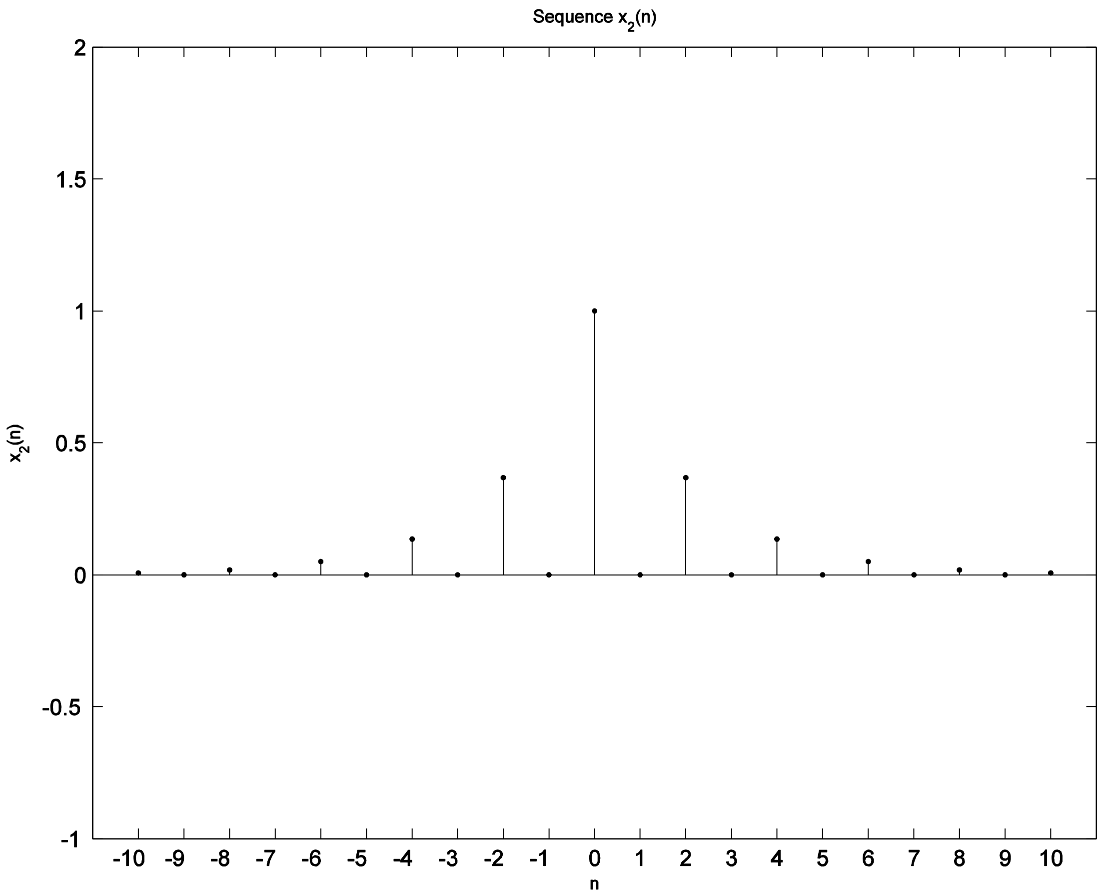


Figure 2.2: Problem P2.1.2 sequence plot

3. $x_3(n) = 10u(n) - 5u(n - 5) - 10u(n - 10) + 5u(n - 15)$.

```
%% P0201c: x3(n) = 10u(n) - 5u(n - 5) + 10u(n - 10) +
% 5u(n - 15).
clc; close all;
x3 = 10*stepseq(0,0,20) - 5*stepseq(5,0,20) -
10*stepseq(10,0,20) ...
+ 5*stepseq(15,0,20);
n3 = [0:20];
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0201c');
Hs = stem(n3,x3,'filled'); set(Hs,'markersize',2);
axis([min(n3)-1,max(n3)+1,min(x3)-1,max(x3)+2]);
ytick = [-6:2:12];
xlabel('n','FontSize',8); ylabel('x_3(n)','FontSize',8);
title('Sequence x_3(n)','FontSize',8);
set(gca,'XTickMode','manual','XTick',n3);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0201c;
```

The plots of $x_3(n)$ is shown in Figure 2.3

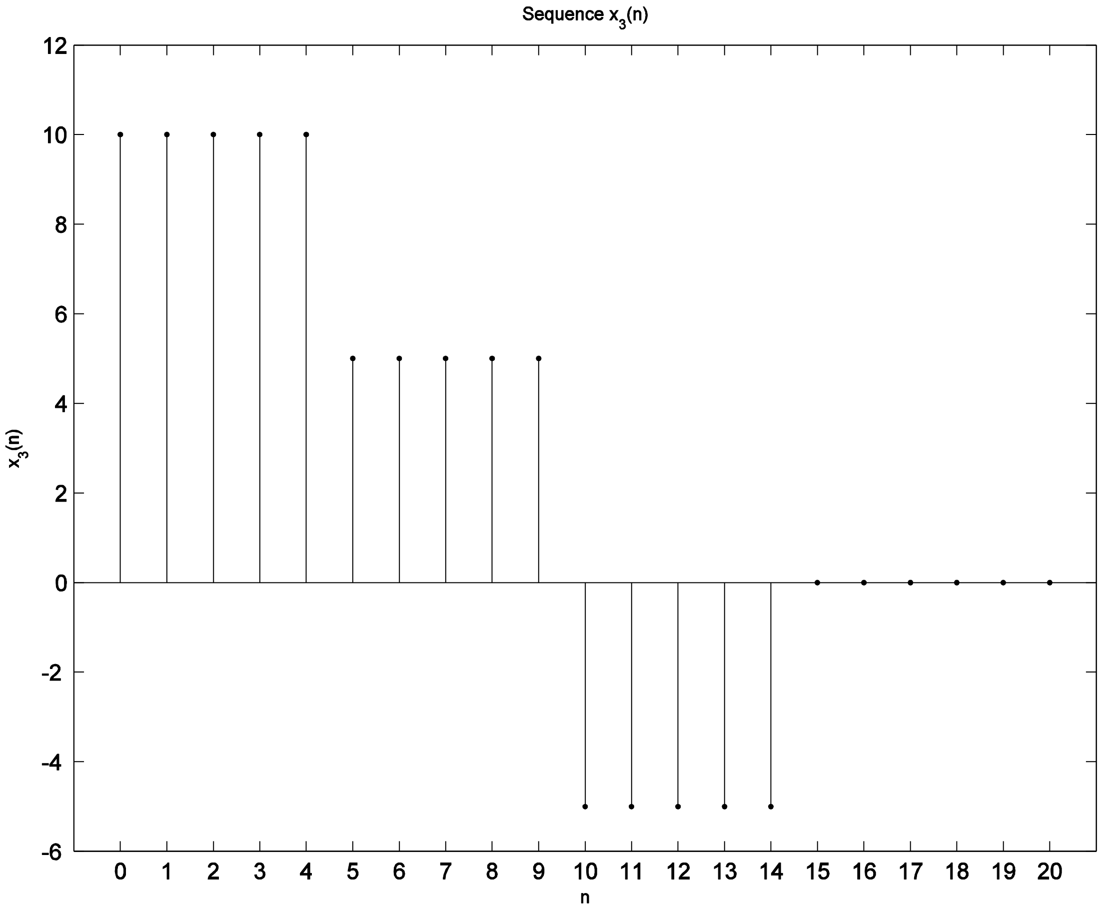


Figure 2.3: Problem P2.1.3 sequence plot

4. $x_4(n) = e^{0.1n}[u(n + 20) - u(n - 10)]$

```
%% P0201d: x4(n) = e ^ {0.1n} [u(n + 20) - u(n - 10)].
clc; close all;
n4 = [-25:15];
x4 = exp(0.1*n4).*(stepseq(-20,-25,15) - stepseq(10,-
25,15));
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0201d');
Hs = stem(n4,x4,'filled'); set(Hs,'markersize',2);
axis([min(n4)-2,max(n4)+2,min(x4)-1,max(x4)+1]);
xlabel('n','FontSize',8); ylabel('x_4(n)','FontSize',8);
title('Sequence x_4(n)','FontSize',8); ntick =
[n4(1):5:n4(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0201d;
```

The plots of $x_4(n)$ is shown in Figure 2.4

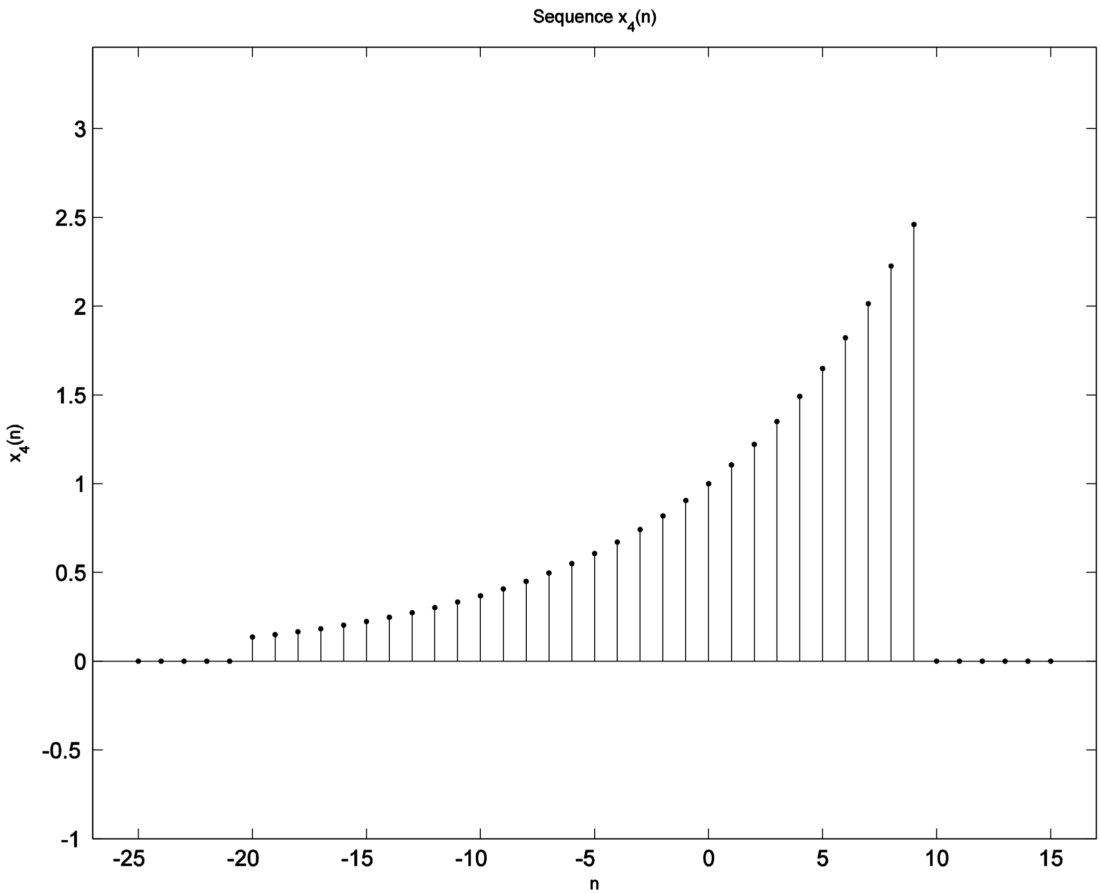


Figure 2.4: Problem P2.1.4 sequence plot

5. $x_5(n) = 5[\cos(0.49\pi n) + \cos(0.51\pi n)]$, $-200 \le n \le 200$. Comment on the waveform shape

```
%% P0201e: x5(n) = 5[cos(0.49*pi*n) + cos(0.51*pi*n)], -
% 200 <= n <= 200.
clc; close all;
n5 = [-200:200]; x5 = 5*(cos(0.49*pi*n5) +
cos(0.51*pi*n5));
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0201e');
Hs = stem(n5,x5,'filled'); set(Hs,'markersize',2);
axis([min(n5)-10,max(n5)+10,min(x5)-2,max(x5)+2]);
xlabel('n','FontSize',8); ylabel('x_5(n)','FOntSize',8);
title('Sequence x_5(n)','FontSize',8);
ntick = [n5(1): 40:n5(end)]; ytick = [-12 -10:5:10 12];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0201e;
```

The plots of $x_5(n)$ is shown in Figure 2.5

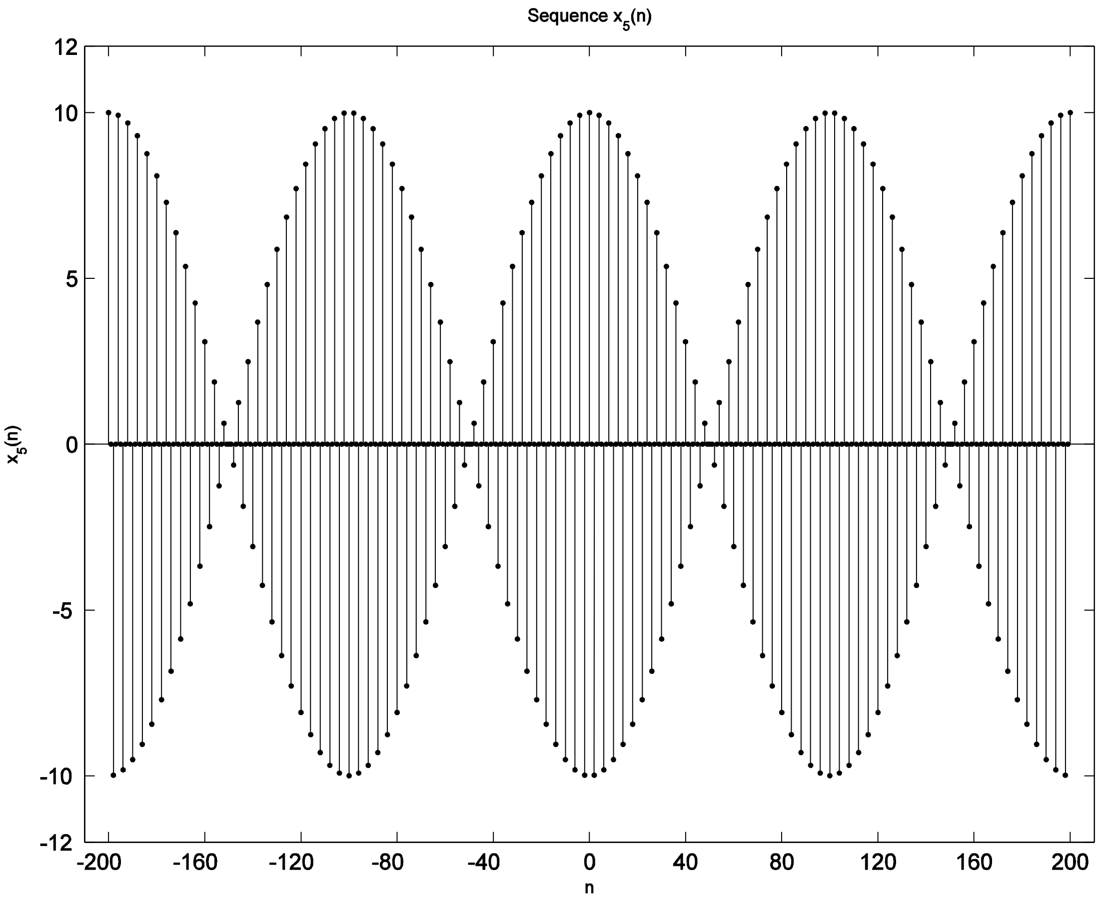


Figure 2.5: Problem P2.1.5 sequence plot

6. $x_6(n) = 2 \sin(0.01\pi n) \cos(0.5\pi n)$, $-200 \leq n \leq 200$

```
%% P0201f: x6(n) = 2 sin(0.01*pi*n) cos(0.5*pi*n), -200
% <= n <= 200.
clc; close all;
n6 = [-200:200]; x6 = 2*sin(0.01*pi*n6).*cos(0.5*pi*n6);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0201f');
Hs = stem(n6,x6,'filled'); set(Hs,'markersize',2);
axis([min(n6)-10,max(n6)+10,min(x6)-1,max(x6)+1]);
xlabel('n','FontSize',8); ylabel('x_6(n)','FontSize',8);
title('Sequence x_6(n)','FontSize',8);
ntick = [n6(1): 40 :n6(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0201f
```

The plots of $x_6(n)$ is shown in Figure 2.6

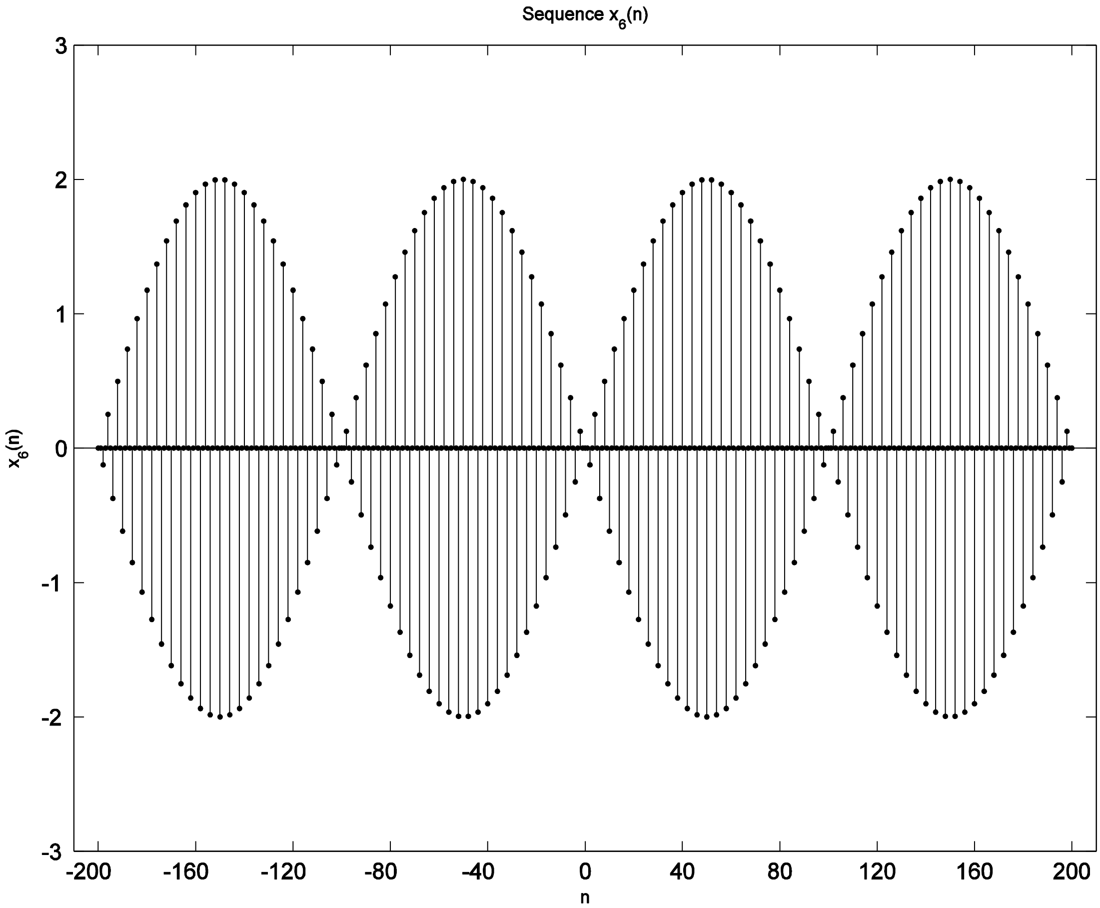


Figure 2.6: Problem P2.1.6 sequence plot

7. $x_7(n) = e^{-0.05n} \sin(0.1\pi n + \pi/3),\ 0 \le n \le 100$

```
%% P0201g: x7(n) = e ^ {-0.05*n}*sin(0.1*pi*n + pi/3), 0
% <= n <=100.
clc; close all;
n7 = [0:100]; x7 = exp(-0.05*n7).*sin(0.1*pi*n7 + pi/3);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0201g');
Hs = stem(n7,x7,'filled'); set(Hs,'markersize',2);
axis([min(n7)-5,max(n7)+5,min(x7)-1,max(x7)+1]);
xlabel('n','FontSize',8); ylabel('x_7(n)','FontSize',8);
title('Sequence x_7(n)','FontSize',8);
ntick = [n7(1): 10:n7(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0201g;
```

The plots of $x_7(n)$ is shown in Figure 2.7

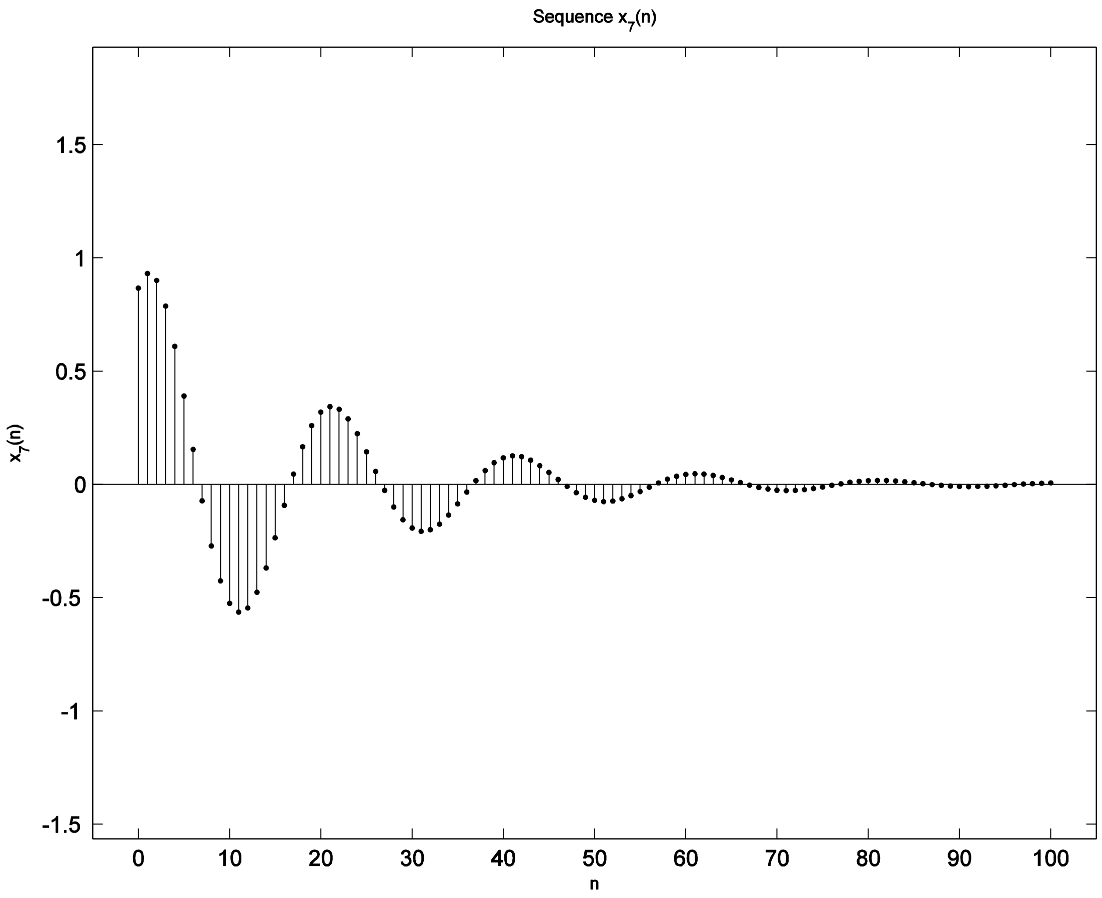


Figure 2.7: Problem P2.1.7 sequence plot

8. $x_8(n) = e^{0.01n}\sin(0.1\pi n)$, $0 \le n \le 100$.

```
%% P0201h: x8(n) = e ^ {0.01*n}*sin(0.1*pi*n), 0 <= n
% <=100.
clc; close all;
n8 = [0:100]; x8 = exp(0.01*n8).*sin(0.1*pi*n8);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0201h');
Hs = stem(n8,x8,'filled'); set(Hs,'markersize',2);
axis([min(n8)-5,max(n8)+5,min(x8)-1,max(x8)+1]);
xlabel('n','FontSize',8); ylabel('x_8(n)','FontSize',8);
title('Sequence x_8(n)','FontSize',8);
ntick = [n8(1): 10:n8(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0201h
```

The plots of $x_8(n)$ is shown in Figure 2.8

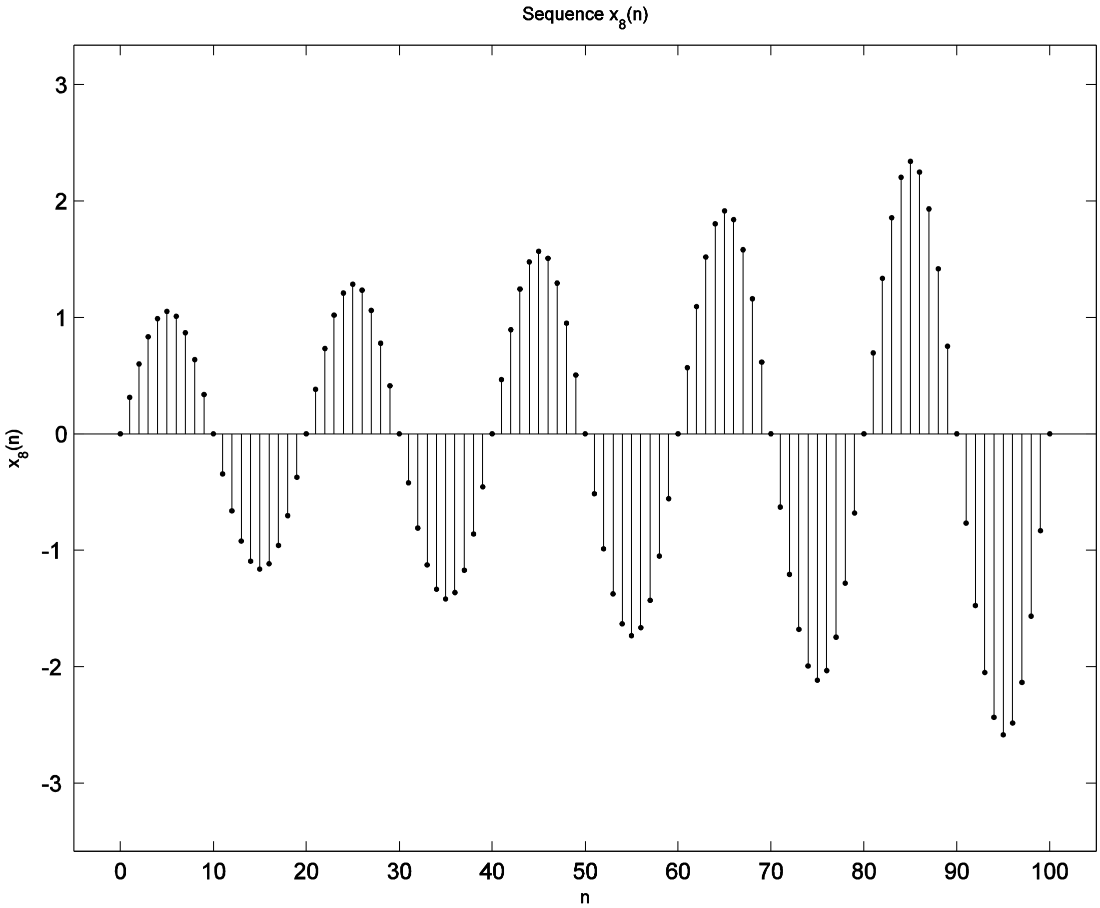


Figure 2.8: Problem P2.1.8 sequence plot

P2.2

Generate the following random sequences and obtain their histogram using the **hist** function with 100 bins. Use the **bar** function to plot each histogram.

1. $x_1(n)$ is a random sequence whose samples are independent and uniformly distributed over [0, 2] interval. Generate 100,000 samples.
2. $x_2(n)$ is a Gaussian random sequence whose samples are independent with mean 10 and variance 10. Generate 10,000 samples.
3. $x_3(n) = x_1(n) + x_1(n - 1)$ where $x_1(n)$ is the random sequence given in part 1 above. Comment on the shape of this histogram and explain the shape.
4. $x_4(n) = \sum_{k=1}^{4} y_k(n)$ where each random sequence $y_k(n)$ is independent of others with samples uniformly distributed over [-0.5, 0.5]. Comment on the shape of this histogram.

Solutions

1. $x_1(n)$ is a random sequence whose samples are independent and uniformly distributed over

[0, 2] interval.
Generate 100,000 samples.

```
% P2.2
%% P0202a: x1(n) = uniform[0,2]
clc; close all;
n1 = [0:100000-1]; x1 = 2*rand(1,100000);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0202a');
[h1,x1out] = hist(x1,100); bar(x1out, h1);
axis([-0.1 2.1 0 1200]);
xlabel('interval','FontSize',8);
ylabel('number of elements','FontSize',8);
title('Histogram of sequence x_1(n) in 100
bins','FontSize',8);
print -deps2 ../EPSFILES/P0202a;
```

The plots of $x_1(n)$ is shown in Figure 2.9.

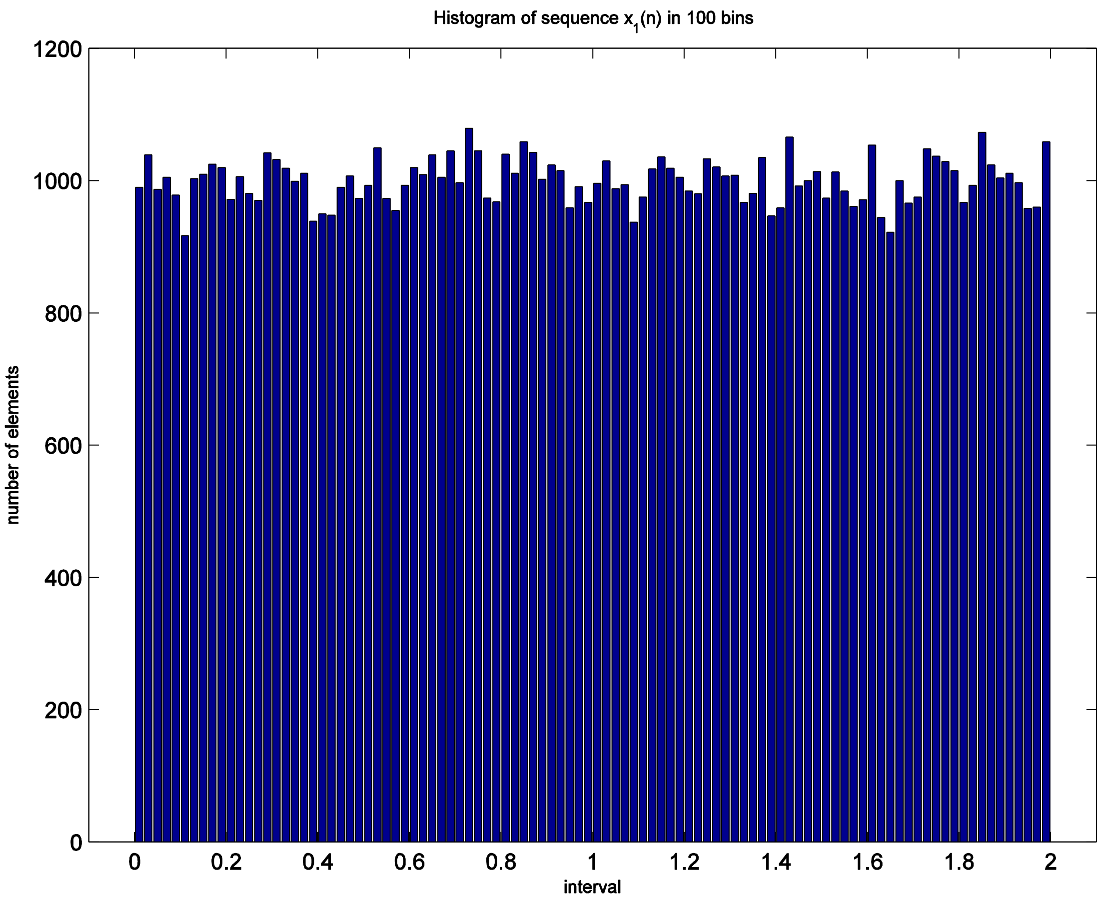


Figure 2.9: Problem P2.2.1 sequence plot

2. $x_2(n)$ is a Gaussian random sequence whose samples are independent with mean 10 and variance 10.

Generate 10,000 samples.

```
%% P0202b: x2(n) = gaussian{10,10}
clc; close all;
n2 = [1:10000]; x2 = 10 + sqrt(10)*randn(1,10000);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0202b');
[h2,x2out] = hist(x2,100); bar(x2out,h2);
xlabel('interval','FontSize',8);
ylabel('number of elements','FontSize',8);
title('Histogram of sequence x_2(n) in 100
bins','FontSize',8);
print -deps2 ../EPSFILES/P0202b;
```

The plots of $x_2(n)$ is shown in Figure 2.10.

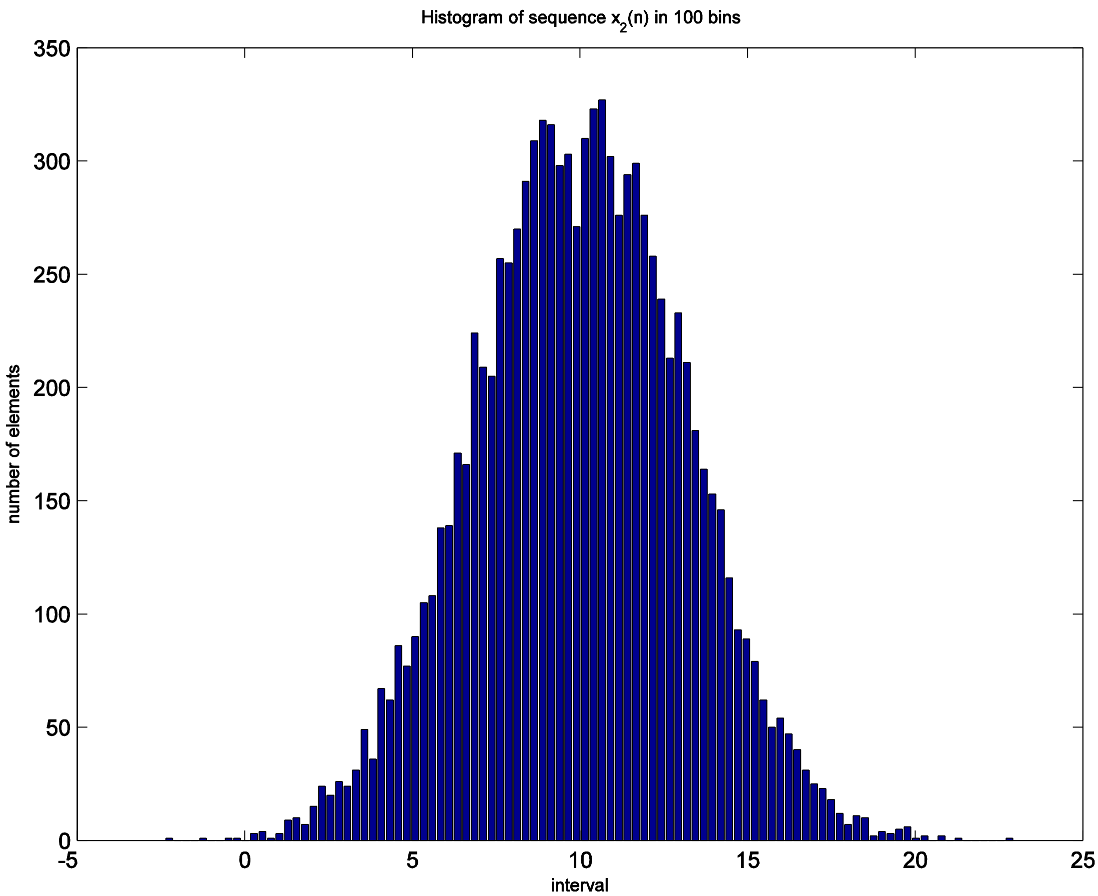


Figure 2.10: Problem P2.2.2 sequence plot

3. $x_3(n) = x_1(n) + x_1(n - 1)$ where $x_1(n)$ is the random sequence given in part 1 above. Comment on the shape of this histogram and explain the shape.

```
%% P0202c: x3(n) = x1(n) + x1(n - 1) where x1(n) =
uniform[0,2]
clc; close all;
n1 = [0:100000-1]; x1 = 2*rand(1,100000);
```

```
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0202c');
[x11,n11] = sigshift(x1,n1,1);
[x3,n3] = sigadd(x1,n1,x11,n11);
[h3,x3out] = hist(x3,100);
bar(x3out,h3); axis([-0.5 4.5 0 2500]);
xlabel('interval','FontSize',8);
ylabel('number of elements','FontSize',8);
title('Histogram of sequence x_3(n) in 100
bins','FontSize',8);
print -deps2 ../EPSFILES/P0202c;
```

The plots of $x_3(n)$ is shown in Figure 2.11.

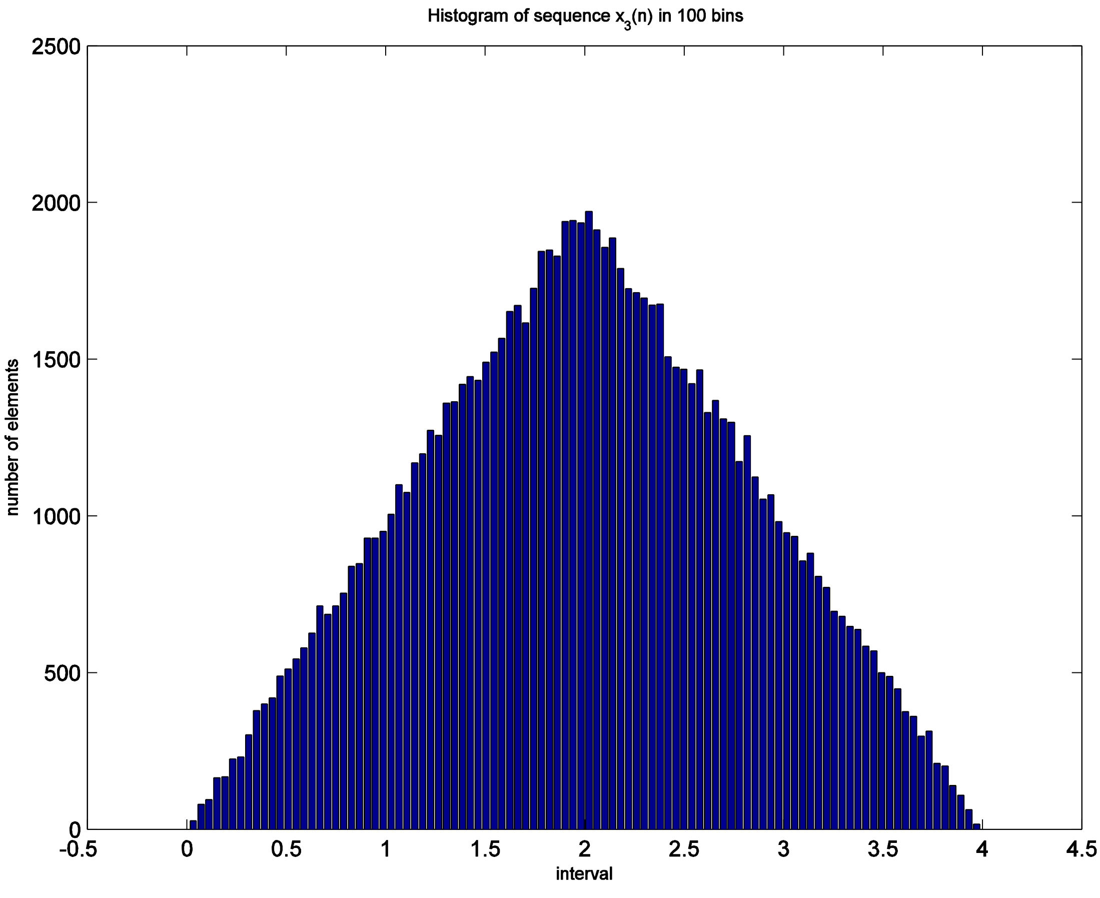


Figure 2.11: Problem P2.2.3 sequence plot

4. $x_4(n) = \sum_{k=1}^{4} y_k(n)$ where each random sequence $y_k(n)$ is independent of others with samples uniformly distributed over [-0.5, 0.5]. Comment on the shape of this histogram.

```
%% P0202d: x4(n) = sum_{k=1} ^ {4} y_k(n), where each
independent of others
% with samples uniformly distributed over [-0.5,0.5];
clc; close all;
y1 = rand(1,100000) - 0.5; y2 = rand(1,100000) - 0.5;
```

```
y3 = rand(1,100000) - 0.5; y4 = rand(1,100000) - 0.5;
x4 = y1 + y2 + y3 + y4;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0202d');
[h4,x4out] = hist(x4,100); bar(x4out,h4);
xlabel('interval','FontSize',8);
ylabel('number of elements','FontSize',8);
title('Histogram of sequence x_4(n) in 100
bins','FontSize',8);
print -deps2 ../EPSFILES/P0202d;
```

The plots of $x_4(n)$ is shown in Figure 2.12.

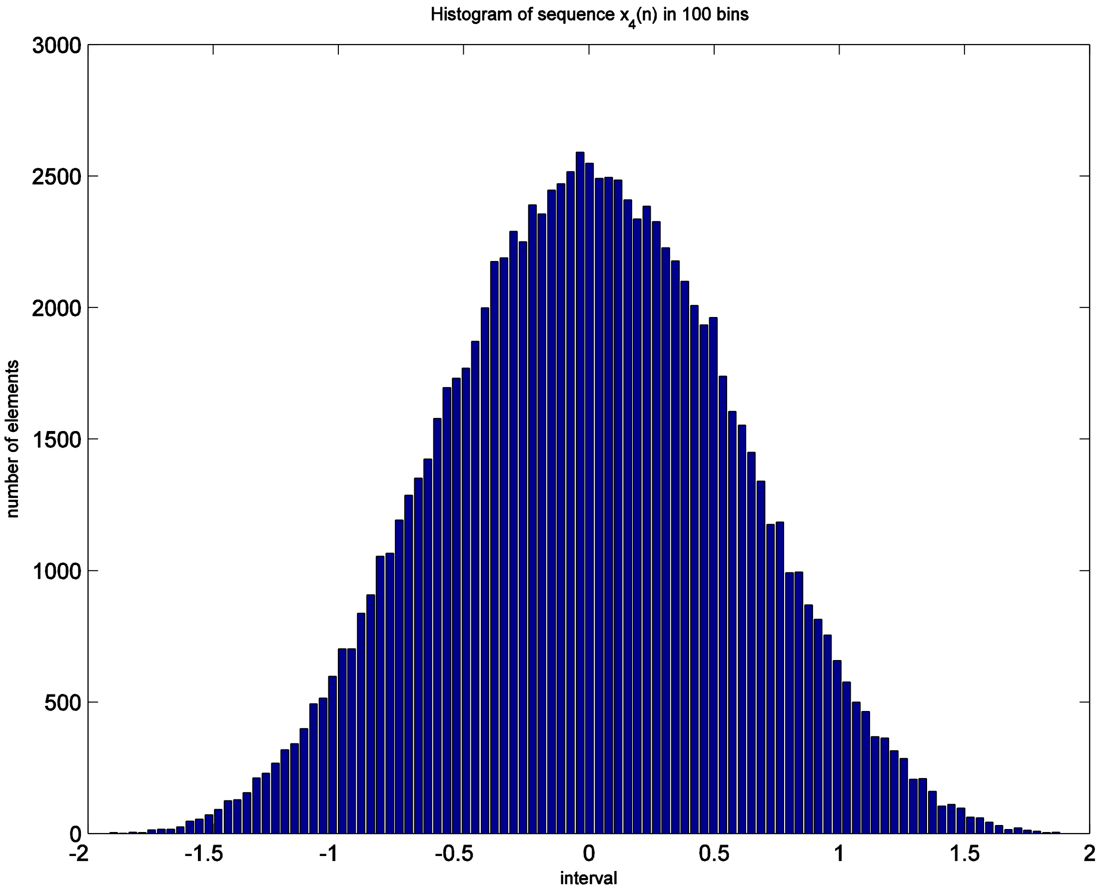


Figure 2.12: Problem P2.2.4 sequence plot

P2.3

Generate the following periodic sequences and plot their samples (using the **stem** function) over the indicated number of periods.

1. $\tilde{x}_1(n) = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$periodic. Plot 5 periods.

$\uparrow$

2. $\tilde{x}_2(n) = e^{0.1n}[u(n) - u(n - 20]$periodic. Plot 3 periods.

3. $\tilde{x}_3(n) = sin(0.1\pi n)[u(n) - u(n - 10)]$. Plot 4 periods.

4. $\tilde{x}_4(n) = \{\ldots, \underset{\uparrow}{1}, 2, 3, \ldots\}$periodic $+ \{\ldots, \underset{\uparrow}{1}, 2, 3, 4, \ldots\}$periodic, $0 \le n \le 24$. What is the period of $\tilde{x}_4(n)$?

Solutions

1. $\tilde{x}_1(n) = \{\ldots, -2, -1, \underset{\uparrow}{0}, 1, 2, \ldots\}_{\text{periodic}}$. Plot 5 periods.

```
%% P0203a: x1(n) = {...,-2,-1,0,1,2,-2,-1,0,1,2...}
periodic. 5 periods
clc; close all;
n1 = [-12:12]; x1 = [-2,-1,0,1,2];
x1 = x1'*ones(1,5); x1 = (x1(:))';
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0203a');
Hs = stem(n1,x1,'filled'); set(Hs,'markersize',2);
axis([min(n1)-1,max(n1)+1,min(x1)-1,max(x1)+1]);
xlabel('n','FontSize',8); ylabel('x_1(n)','FontSize',8);
title('Sequence x_1(n)','FontSize',8);
ntick = [n1(1):2:n1(end)]; ytick = [min(x1) - 1:max(x1) +
1];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0203a
```

The plots of $x_1(n)$ is shown in Figure 2.13.

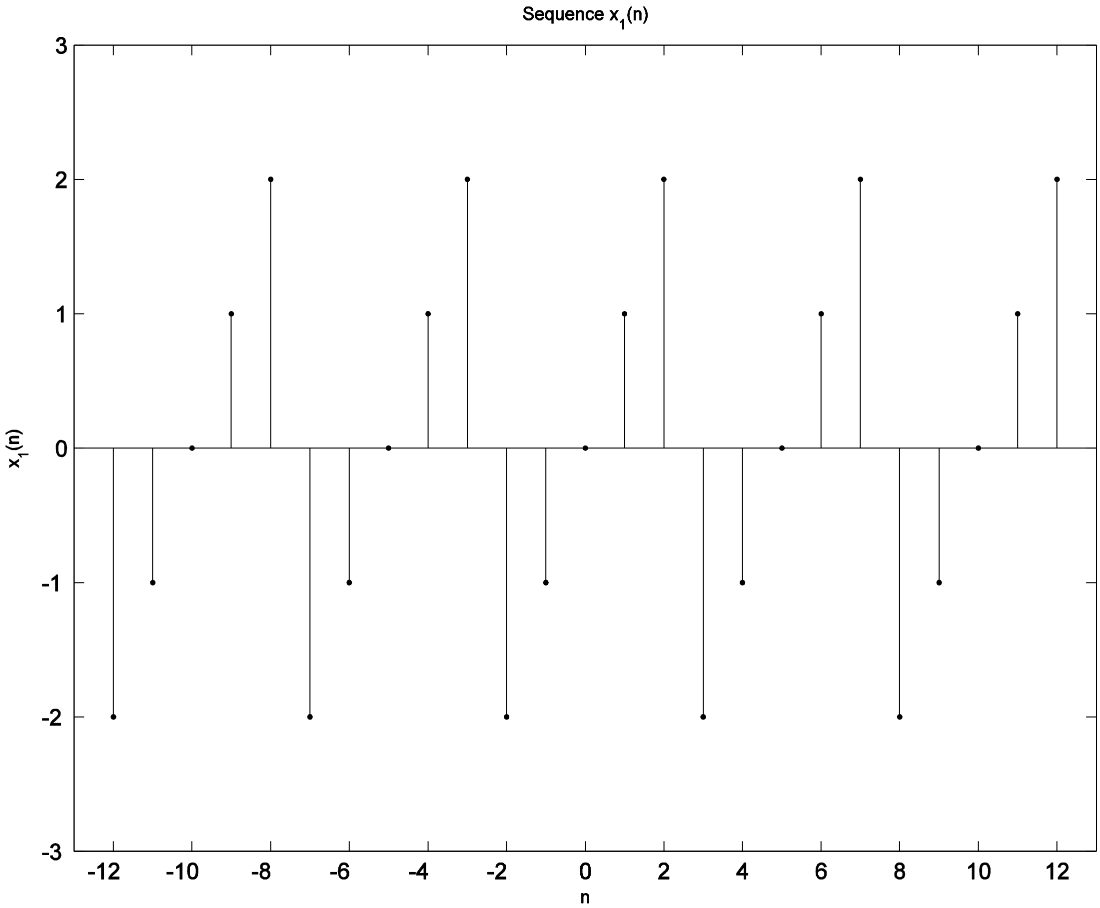


Figure 2.13: Problem P2.3.1 sequence plot

2. $\tilde{x}_2(n) = e^{0.1n}[u(n) - u(n-20)]_{\text{periodic}}$. Plot 3 periods.

```
%% P0203b: x2 = e ^ {0.1n} [u(n) - u(n-20)] periodic. 3
periods
clc; close all;
n2 = [0:21]; x2 = exp(0.1*n2).*(stepseq(0,0,21)-
stepseq(20,0,21));
x2 = x2'*ones(1,3); x2 = (x2(:))'; n2 = [-22:43];
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0203b');
Hs = stem(n2,x2,'filled'); set(Hs,'markersize',2);
axis([min(n2)-2,max(n2)+4,min(x2)-1,max(x2)+1]);
xlabel('n','FontSize',8); ylabel('x_2(n)','FontSize',8);
title('Sequence x_2(n)','FontSize',8);
ntick = [n2(1):4:n2(end)-5 n2(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0203b;
```

The plots of $x_3(n)$ is shown in Figure 2.14.

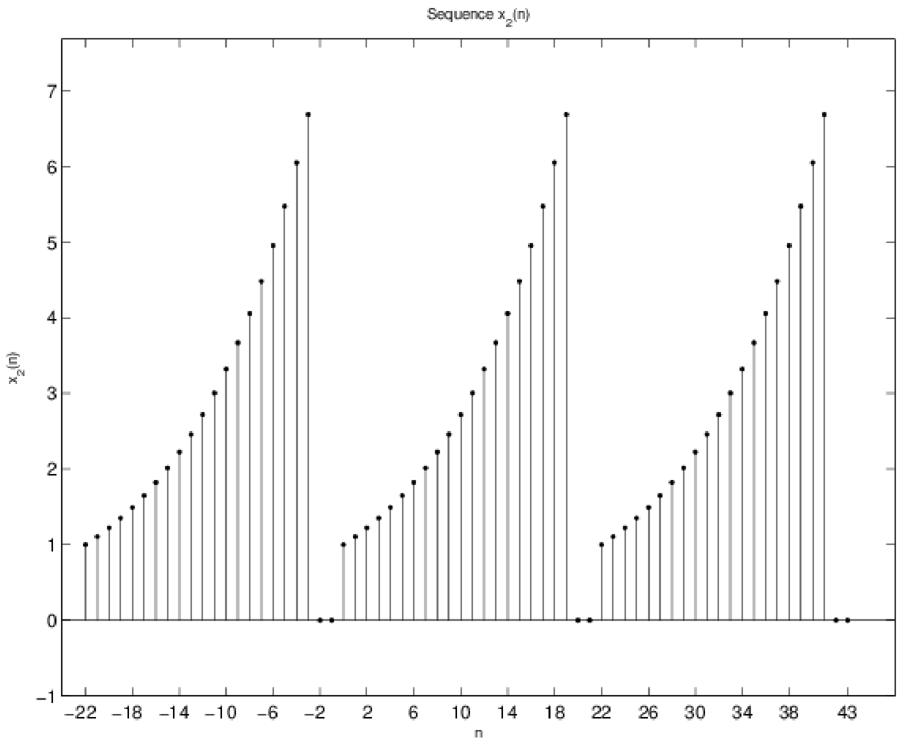


Figure 2.14: Problem P2.3.2 sequence plot

3. $\tilde{x}_3(n) = sin(0.1\pi n)[u(n) - u(n-10)]$. Plot 4 periods.

```
%% P0203c: x1(n) = {...,-2,-1,0,1,2,-2,-1,0,1,2...}
periodic. 4 periods
clc; close all;
n3 = [0:11]; x3 = sin(0.1*pi*n3).*(stepseq(0,0,11)-
stepseq(10,0,11));
x3 = x3'*ones(1,4); x3 = (x3(:))'; n3 = [-12:35];
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0203c');
Hs = stem(n3,x3,'filled'); set(Hs,'markersize',2);
axis([min(n3)-1,max(n3)+1,min(x3)-0.5,max(x3)+0.5]);
xlabel('n','FontSize',8); ylabel('x_3(n)','FontSize',8);
title('Sequence x_3(n)','FontSize',8);
ntick = [n3(1):4:n3(end)-3 n3(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0203c;
```

The plots of $x_3(n)$ is shown in Figure 2.15.

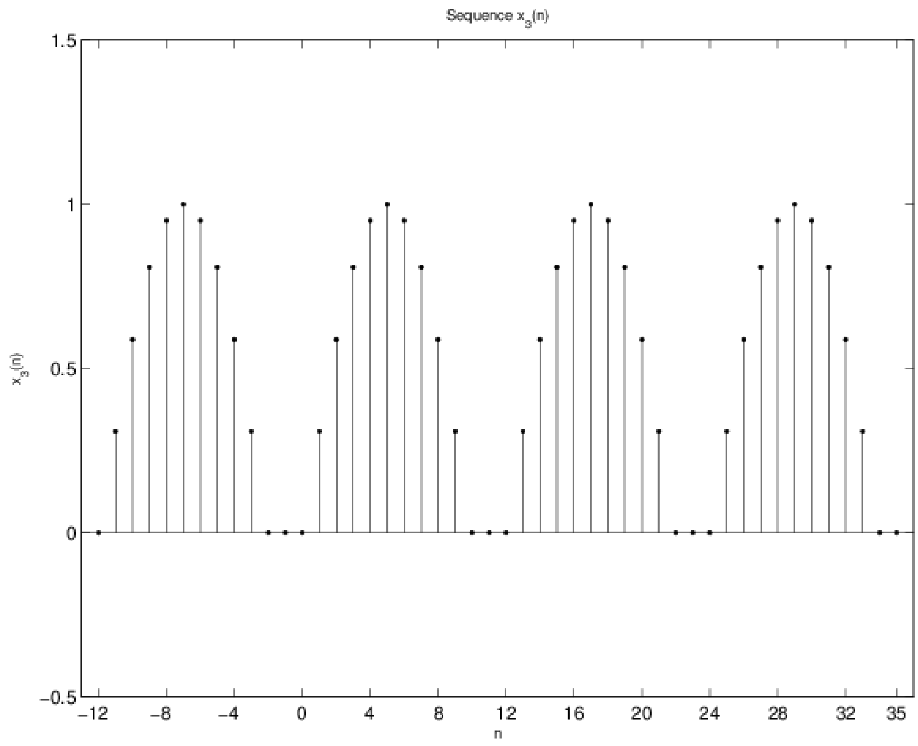


Figure 2.15: Problem P2.3.3 sequence plot

4. $\tilde{x}_4(n) = \{\ldots, \underset{\uparrow}{1}, 2, 3, \ldots\}_{\text{periodic}} + \{\ldots, \underset{\uparrow}{1}, 2, 3, 4, \ldots\}_{\text{periodic}},\ 0 \le n \le 24$. What is the period of $\tilde{x}_4(n)$?

```
%% P0203d x1(n) = {...,-2,-1,0,1,2,-2,-1,0,1,2...}
periodic. 5 periods
clc; close all;
n4 = [0:24]; x4a = [1 2 3]; x4a = x4a'*ones(1,9); x4a =
(x4a(:))';
% There is tatolly 25 pionts between 0 and 24 and
therefore x4a and x4b both need another more
% period.
x4b = [1 2 3 4]; x4b = x4b'*ones(1,7); x4b = (x4b(:))';
x4 = x4a(1:25) + x4b(1:25);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0203d');
Hs = stem(n4,x4,'filled'); set(Hs,'markersize',2);
axis([min(n4)-1,max(n4)+1,min(x4)-1,max(x4)+1]);
xlabel('n', 'FontSize', 8);
ylabel('x_4(n)','FontSize',8);
title('Sequence x_4(n):Period = 12','FontSize',8);
ntick = [n4(1) :2:n4(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0203d;
```

The plots of $x_4(n)$ is shown in Figure 2.16.

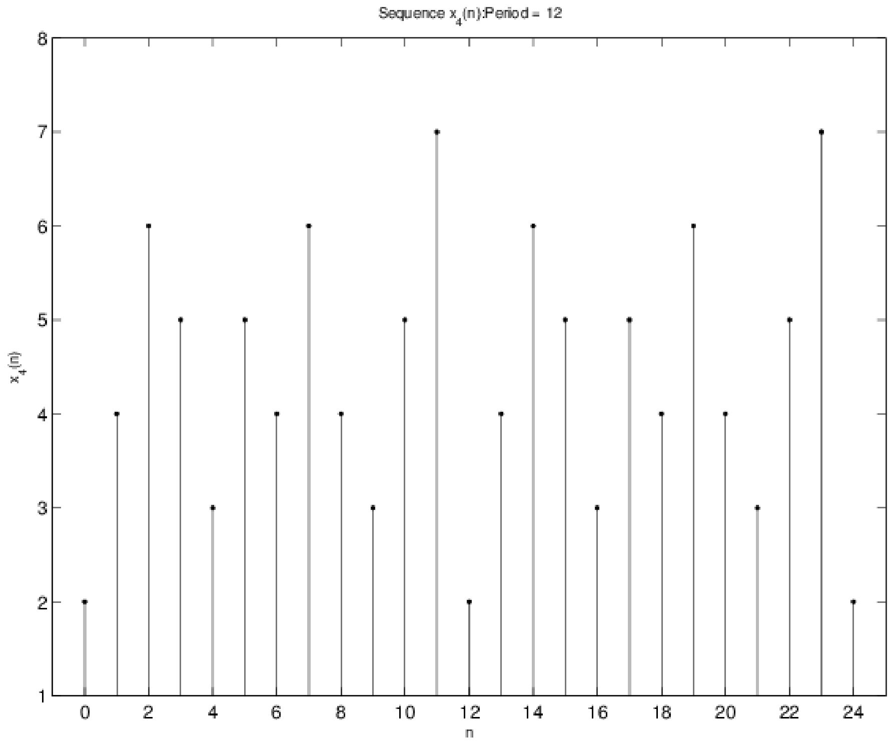


Figure 2.16: Problem P2.3.4 sequence plot

P2.4

Let x(n) = {2,4, -3, 1, -5, 4, 7}. Generate and plot the samples (use the **stem** function) of
↑
the following sequences.

1. $x_1(n) = 2x(n - 3) + 3x(n + 4) - x(n)$
2. $x_2\ (n) = 4x(4 + n) + 5x(n + 5) + 2x(n)$
3. $x_3\ (n) = x(n + 3)x(n - 2) + x(1 - n)x(n + 1)$
4. $x_4\ (n) = 2e^{0.5n}x(n) + \cos(0.1\pi n)\ x(n + 2)\ ,\ -10 \le n \le 10$

Solutions

1. $x_1(n) = 2x(n - 3) + 3x(n + 4) - x(n)$

```
% P2.4
%% P0204a: x(n) = [2,4,-3,1,-5,4,7]; -3 <=n <= 3;
% x1(n) = 2x(n - 3) + 3x(n + 4) - x(n)
clc; close all;
n = [-3:3]; x = [2,4,-3,1,-5,4,7];
[x11,n11] = sigshift(x,n,3); % shift by 3
[x12,n12] = sigshift(x,n,-4); % shift by -4
[x13,n13] = sigadd(2*x11,n11,3*x12,n12); % add two
sequences
```

```
[x1,n1] = sigadd(x13,n13,-x,n); % add two sequences
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0204a');
Hs = stem(n1,x1,'filled'); set(Hs,'markersize',2);
axis([min(n1)-1,max(n1)+1,min(x1)-3,max(x1)+1]);
xlabel('n','FontSize',8);
ylabel('x_1(n)','FontSize',8);
title('Sequence x_1(n)','FontSize',8); ntick = n1;
ytick = [min(x1)-3:5:max(x1)+1];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0204a;
```

The plots of $x_1(n)$ is shown in Figure 2.17.

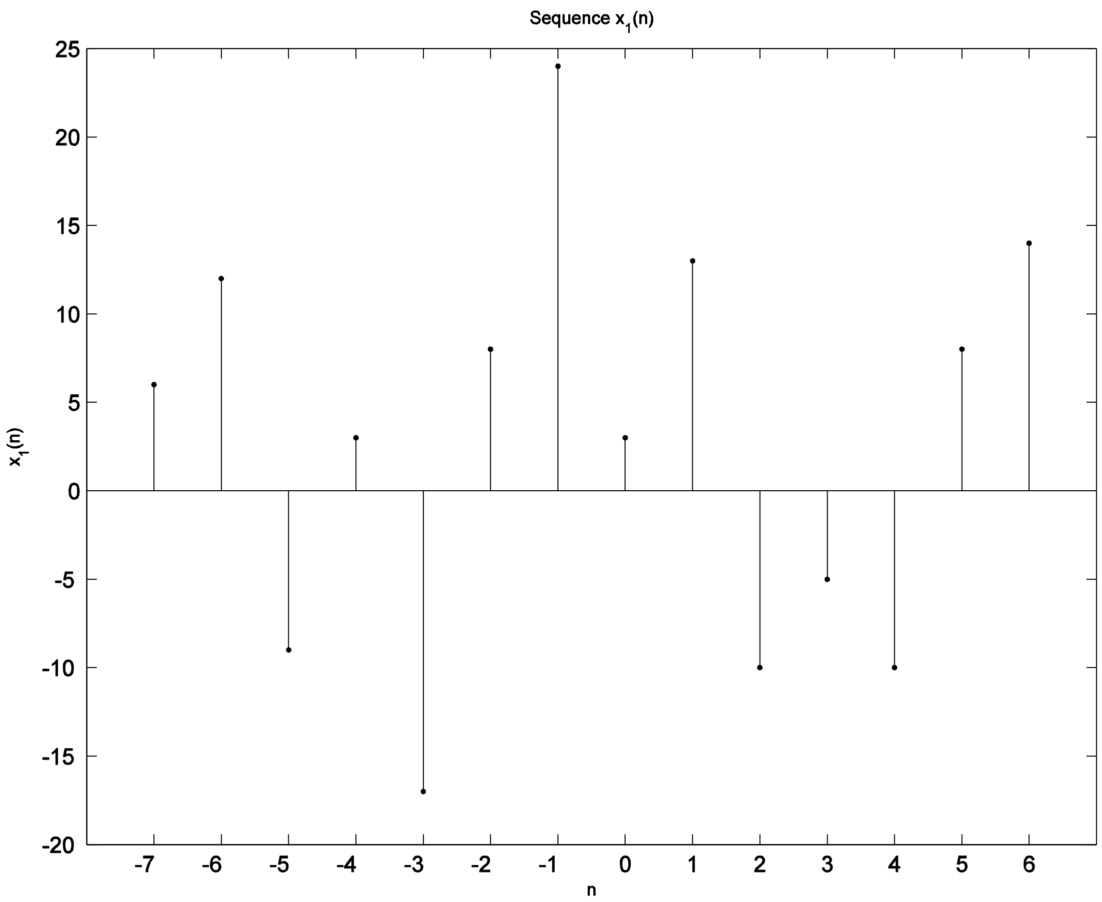


Figure 2.17: Problem P2.4.1 sequence plot

2. $x_2(n) = 4x(4+n) + 5x(n+5) + 2x(n)$

```
%% P0204b: x(n) = [2,4,-3,1,-5,4,7]; -3 <=n <= 3;
% x2(n) = 4x(4 + n) + 5x(n + 5) + 2x(n)
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0204b');
n = [-3:3]; x = [2,4,-3,1,-5,4,7];
```

```
[x21,n21] = sigshift(x,n,-4); % shift by -4
[x22,n22] = sigshift(x,n,-5); % shift by -5
[x23,n23] = sigadd(4*x21,n21,5*x22,n22); % add two
sequences
[x2,n2] = sigadd(x23,n23,2*x,n); % add two sequences
Hs = stem(n2,x2,'filled'); set(Hs,'markersize',2);
axis([min(n2)-1,max(n2)+1,min(x2)-4,max(x2)+6]);
xlabel('n','FontSize',8); ylabel('x_2(n)','FontSize',8);
title('Sequence x_2(n)','FontSize',8); ntick = n2;
ytick = [-25 -20:10:60 65];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0204b;
```

The plots of $x_2(n)$ is shown in Figure 2.18.

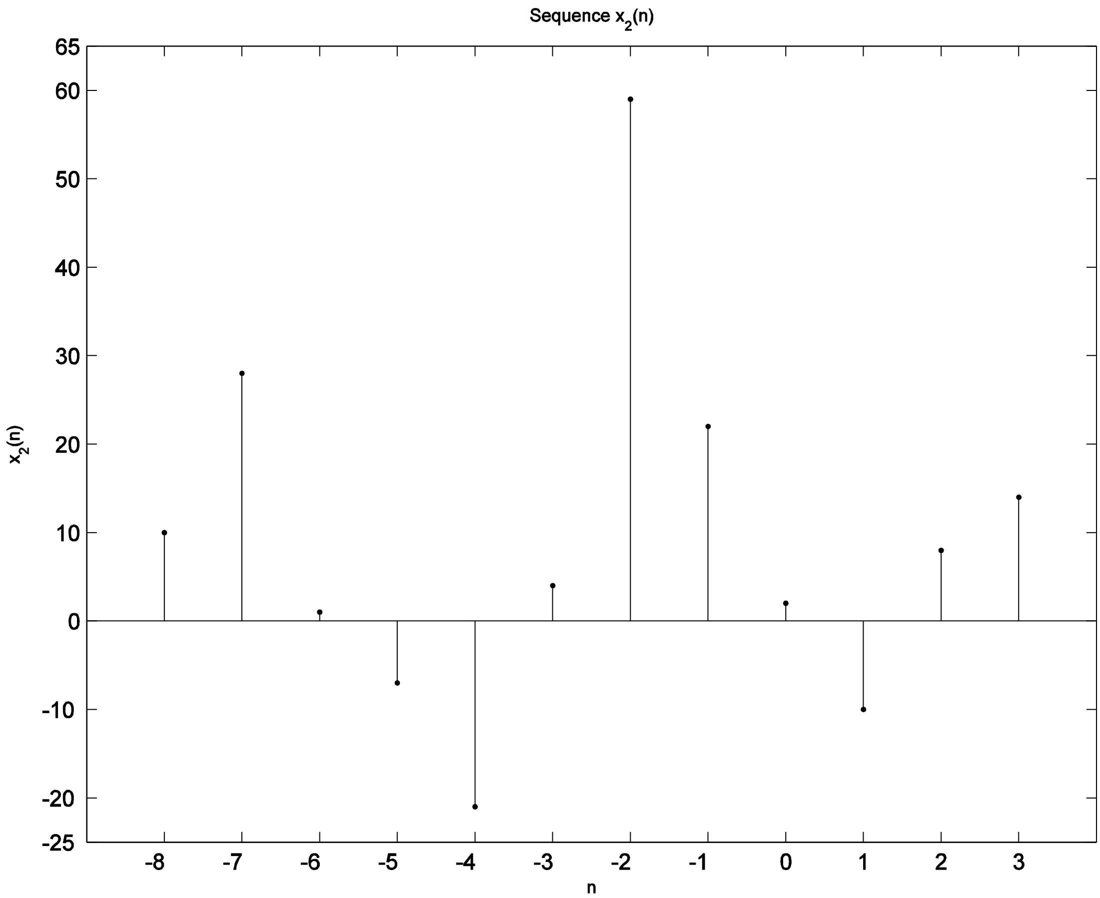


Figure 2.18: Problem P2.4.2 sequence plot

3. $x_3(n) = x(n + 3)x(n - 2) + x(1 - n)x(n + 1)$

```
%% P204c: x(n) = [2,4,-3,1,-5,4,7]; -3 <=n <= 3;
% x3(n) = x(n + 3)x(n - 2) + x(1 - n)x(n + 1)
clc; close all;
n = [-3:3]; x = [2,4,-3,1,-5,4,7]; % given sequence x(n)
[x31,n31] = sigshift(x,n,-3); % shift sequence by -3
```

```
[x32,n32] = sigshift(x,n,2); % shift sequence by 2
[x33,n33] = sigmult(x31,n31,x32,n32); % multiply 2
sequences
[x34,n34] = sigfold(x,n); % fold x(n)
[x34,n34] = sigshift(x34,n34,1); % shift x(-n) by 1
[x35,n35] = sigshift(x,n,-1); % shift x(n) by -1
[x36,n36] = sigmult(x34,n34,x35,n35); % multiply 2
sequences
[x3,n3] = sigadd(x33,n33,x36,n36); % add 2 sequences
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0204c');
Hs = stem(n3,x3,'filled'); set(Hs,'markersize',2);
axis([min(n3)-1,max(n3)+1,min(x3)-10,max(x3)+10]);
xlabel('n','FontSize',8); ylabel('x_3(n)','FontSize',8);
title('Sequence x_3(n)','FontSize',8);
ntick = n3; ytick = [-30:10:60];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0204c;
```

The plots of $x_3(n)$ is shown in Figure 2.19.

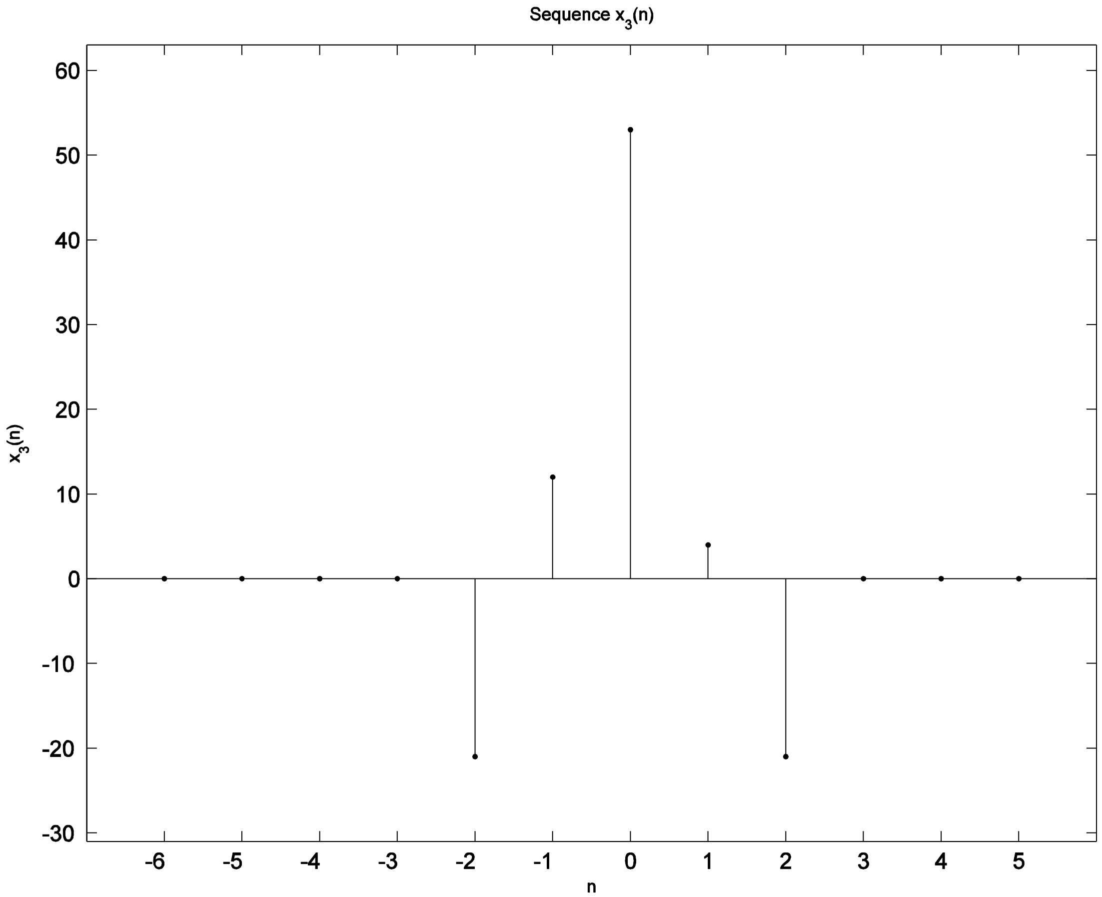


Figure 2.19: Problem P2.4.3 sequence plot

4. $x_4(n) = 2e^{0.5n} x(n) + \cos(0.1\pi n)\, x(n+2),\ -10 \le n \le 10$

```
%% P0204d: x(n) = [2,4,-3,1,-5,4,7]; -3 <=n <= 3;
% x4(n) = 2*e^{0.5n}*x(n)+cos(0.1*pi*n)*x(n+2), -10 <=n<
=10
clc; close all;
n = [-3:3]; x = [2,4,-3,1,-5,4,7]; % given sequence x(n)
n4 = [-10:10]; x41 = 2*exp(0.5*n4); x412 =
cos(0.1*pi*n4);
[x42,n42] = sigmult(x41,n4,x,n);
[x43,n43] = sigshift(x,n,-2);
[x44,n44] = sigmult(x412,n42,x43,n43);
[x4,n4] = sigadd(x42,n42,x44,n44);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0204d');
Hs = stem(n4,x4,'filled'); set(Hs,'markersize',2);
axis([min(n4)-1,max(n4)+1,min(x4)-11,max(x4)+10]);
xlabel('n','FontSize',8); ylabel('x_4(n)','FontSize',8);
title('Sequence x_4(n)','FontSize',8);
ntick = n4; ytick = [ -20:10:70];
set(gca,'XTickMode','manual','XTick',ntick);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0204d;
```

The plots of $x_4(n)$ is shown in Figure 2.20.

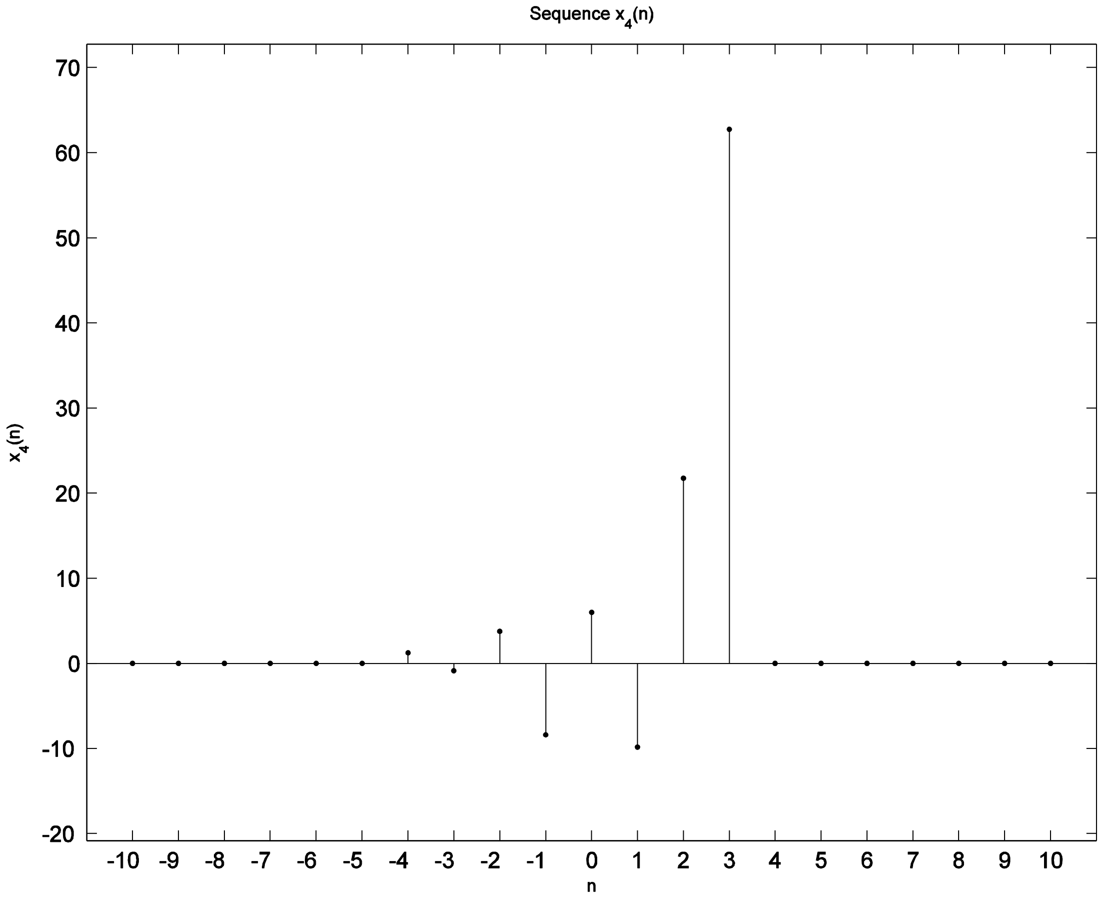


Figure 2.20: Problem P2.4.4 sequence plot

P2.5

The complex exponential sequence $e^{jw_0 n}$ or the sinusoidal sequence $\cos(\omega_0 n)$ are periodic if the *normalized* frequency $f0 \triangleq \frac{\omega 0}{2\pi}$ is a rational number; that is, $f0 \triangleq \frac{K}{N}$, where K and N are integers.

1. Prove the above result.
2. Generate $\exp(0.1\pi n)$, $-100 \le n \le 100$. Plot its real and imaginary parts using the **stem** function. Is this sequence periodic? If it is, what is its fundamental period? From the examination of the plot what interpretation can you give to the integers K and N above?
3. Generate and plot $\cos(0.1n)$, $-20 \le n \le 20$. Is this sequence periodic? What do you conclude from the plot? If necessary examine the values of the sequence in MATLAB to arrive at your answer.

Solutions

1. Analytical proof: The exponential sequence is periodic if

$$e^{j2\pi f_0(n+N)} = e^{j2\pi f_0 n} \text{ or } e^{j2\pi f_0 N} = 1 \Rightarrow f_0 N = K \text{ (an integer)}$$

which proves the result.

2. $x_1 = \exp(0.1\pi n)$, $-100 \le n \le 100$.

```
% P2.5
%% P0205b: x1(n) = e^{0.1*j*pi*n} -100 <=n <=100
clc; close all;
n1 = [-100:100]; x1 = exp(0.1*j*pi*n1);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0205b');
subplot(2,1,1); Hs1 = stem(n1,real(x1),'filled');
set(Hs1,'markersize',2);
axis([min(n1)-5,max(n1)+5,min(real(x1))-
1,max(real(x1))+1]);
xlabel('n','FontSize',8);
ylabel('Real(x_1(n))','FontSize',8);
title(['Real part of sequence x_1(n) = ' ...
'exp(0.1 \times j \times pi \times n)' char(10) ...
' Period = 20, K = 1, N = 20'],'FontSize',8);
ntick = [n1(1):20:n1(end)];
set(gca,'XTickMode','manual','XTick',ntick);
subplot(2,1,2); Hs2 = stem(n1,imag(x1),'filled');
set(Hs2,'markersize',2);
axis([min(n1)-5,max(n1)+5,min(real(x1))-
1,max(real(x1))+1]);
xlabel('n','FontSize',8);
ylabel('Imag(x_1(n))','FontSize',8);
title(['Imaginary part of sequence x_1(n) = ' ...
'exp(0.1 \times j \times pi \times n) ' char(10) ...
' Period = 20, K = 1, N = 20'],'FontSize',8);
ntick = [n1(1):20:n1(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0205b;
```

The plots of $x_1(n)$ is shown in Figure 2.21. Since $f_0 = 0.1/2 = 1/20$ the sequence is periodic. From the plot in Figure 2.21 we see that in one period of 20 samples $x_1(n)$ exhibits cycle. This is true whenever K and N are relatively prime

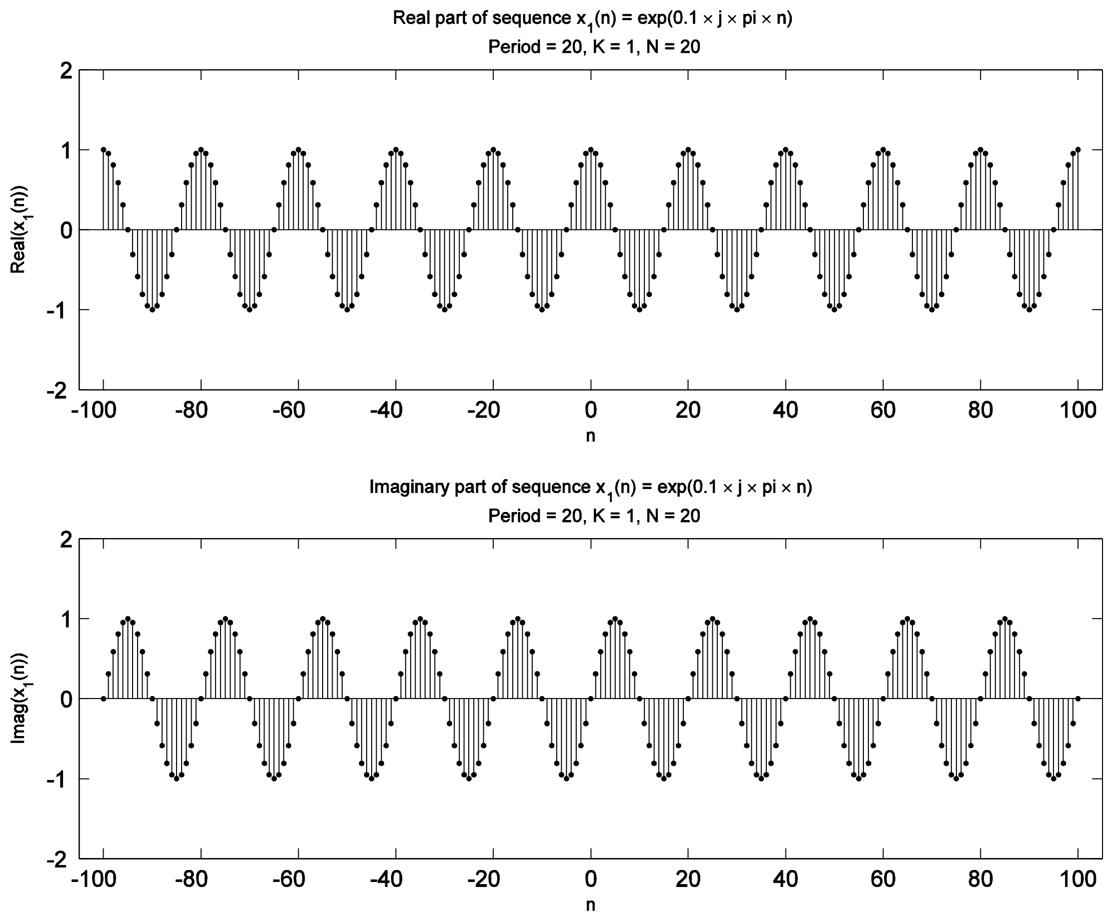


Figure 2.21: Problem P2.5.2 sequence plot

3. $x_2 = \cos(0.1n), -20 \leq n \leq 20$.

```
%% P0205c: x2(n) = cos(0.1n), -20 <= n <= 20
clc; close all;
n2 = [-20:20]; x2 = cos(0.1*n2);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0205c');
Hs = stem(n2,x2,'filled'); set(Hs,'markersize',2);
axis([min(n2)-1,max(n2)+1,min(x2)-1,max(x2)+1]);
xlabel('n','FontSize',8); ylabel('x_2(n)','FontSize',8);
title(['Sequence x_2(n) = cos(0.1 \times n)' char(10) ...
'Not periodic since f_0 = 0.1 / (2 \times \pi)' ...
' is not a rational number'], 'FontSize',8);
ntick = [n2(1):4:n2(end)];
set(gca,'XTickMode','manual','XTick',ntick);
print -deps2 ../EPSFILES/P0205c;
```

The plots of $x_1(n)$ is shown in Figure 2.22. In this case f_0 is not a rational number and hence the sequence $x_2(n)$ is not periodic. This can be clearly seen from the plot of $x_2(n)$ in Figure 2.22.

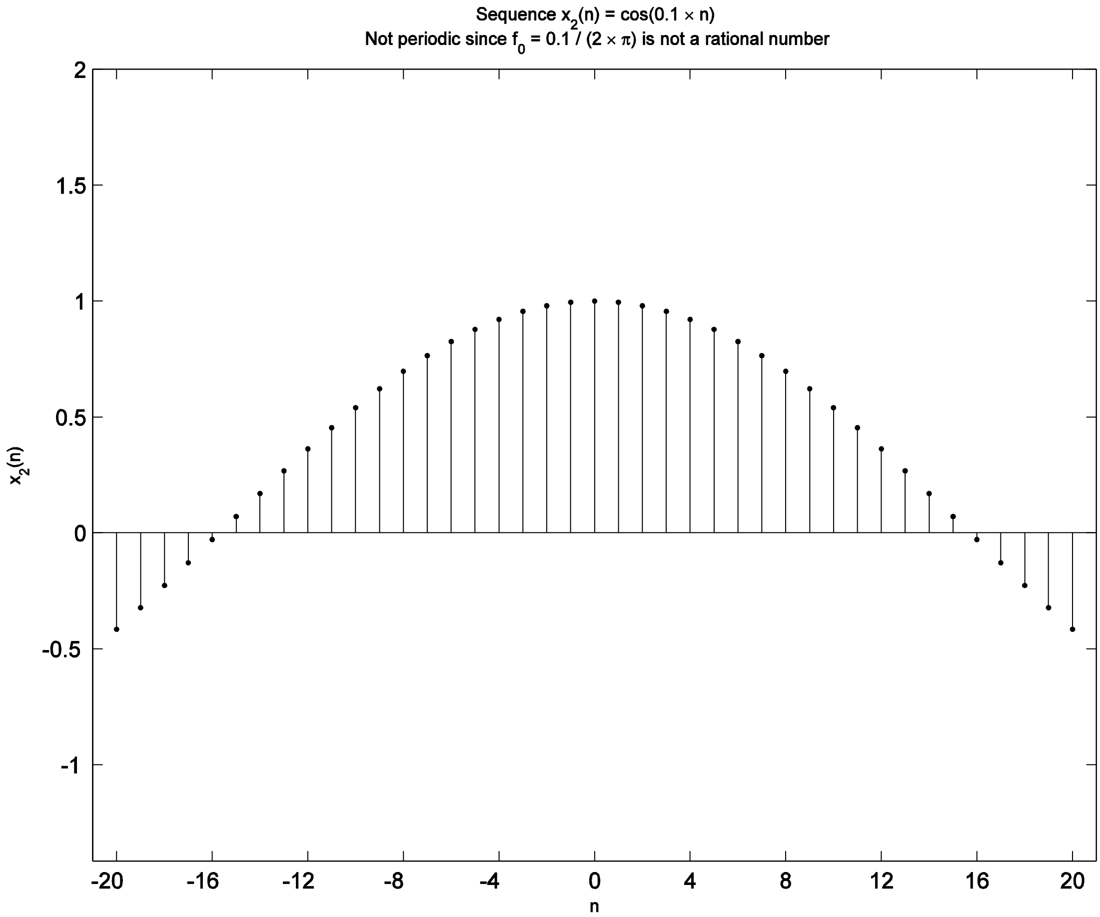


Figure 2.22: Problem P2.5.2 sequence plot

P2.6

Using the **evenodd** function, decompose the following sequences into their even and odd components. Plot these components using the **stem** function.

1. $x_1(n) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
   $\uparrow$
2. $x_2(n) = e^{0.1n}[u(n + 5) - u(n - 10)]$.
3. $x_2(n) = \cos(0.2\pi n + \pi/4), -20 \le n \le 20$.
4. $x_4(n) = e^{-0.05n} \sin(0.1\pi n + \pi/3), 0 \le n \le 100$.

Solutions

1. $x_1(n) = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.
   $\uparrow$

```
% P2.6
%% P0206a: % Even odd decomposition of x1(n) = [0 1 2 3 4
5 6 7 8 9];
```

```
% n = 0:9;
clc; close all;
x1 = [0 1 2 3 4 5 6 7 8 9]; n1 = [0:9]; [xe1,xo1,m1] =
evenodd(x1,n1);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0206a');
subplot(2,1,1); Hs = stem(m1,xe1,'filled');
set(Hs,'markersize',2);
axis([min(m1)-1,max(m1)+1,min(xe1)-1,max(xe1)+1]);
xlabel('n','FontSize',8); ylabel('x_e(n)','FontSize',8);
title('Even part of x_1(n)','FontSize',8);
ntick = [m1(1):m1(end)]; ytick = [-1:5];
set(gca,'XTick',ntick);set(gca,'YTick',ytick);
subplot(2,1,2); Hs = stem(m1,xo1,'filled');
set(Hs,'markersize',2);
axis([min(m1)-1,max(m1)+1,min(xo1)-2,max(xo1)+2]);
xlabel('n','FontSize',8); ylabel('x_o(n)','FontSize',8);
title('Odd part of x_1(n)','FontSize',8);
ntick = [m1(1):m1(end)]; ytick = [-6:2:6];
set(gca,'XTick',ntick);set(gca,'YTick',ytick);
print -deps2 ../EPSFILES/P0206a;
```

The plots of $x_1(n)$ is shown in Figure 2.23.

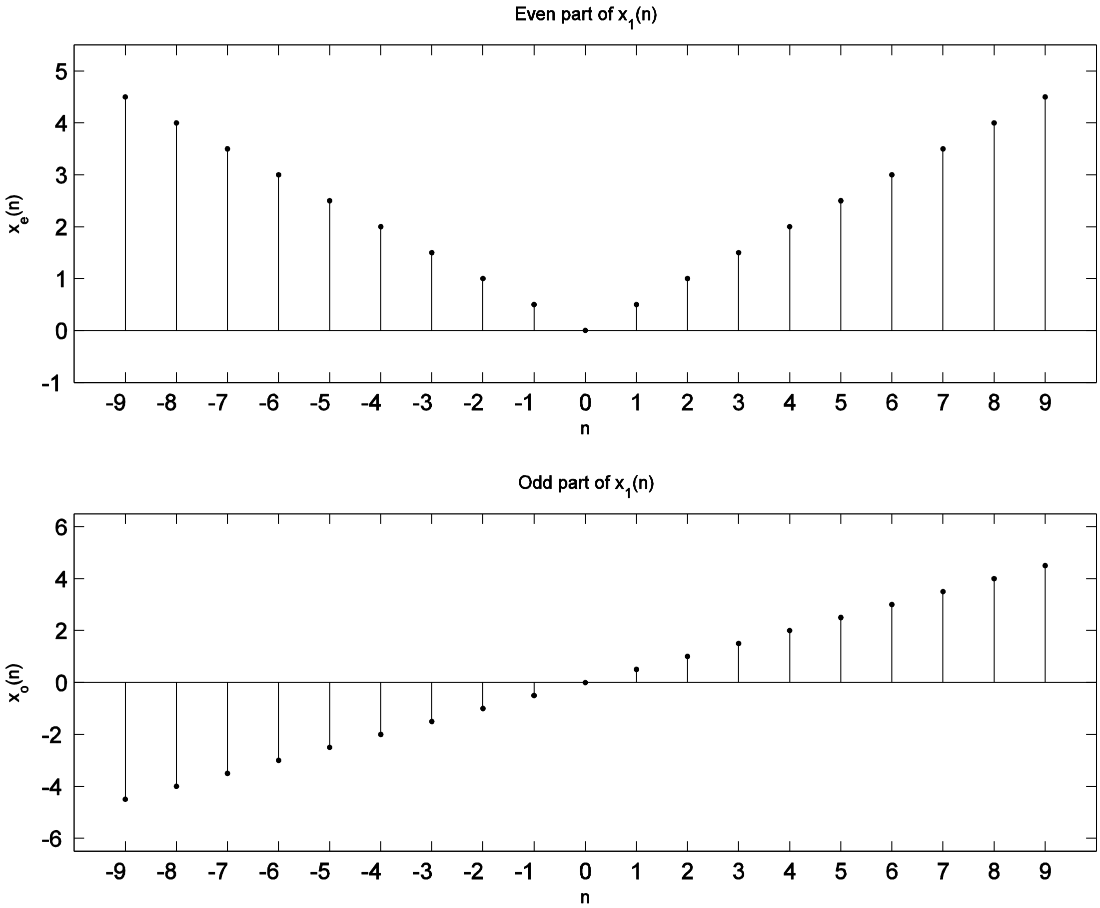


Figure 2.23: Problem P2.6.1 sequence plot

2. $x_2(n) = e^{0.1n}[u(n + 5) - u(n - 10)]$.

```
%% P0206b: Even odd decomposition of x2(n) = e ^ {0.1n}
[u(n + 5) - u(n - 10)];
clc; close all;
n2 = [-8:12]; x2 = exp(0.1*n2).*(stepseq(-5,-8,12) -
stepseq(10,-8,12));
[xe2,xo2,m2] = evenodd(x2,n2);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0206b');
subplot(2,1,1); Hs = stem(m2,xe2,'filled');
set(Hs,'markersize',2);
axis([min(m2)-1,max(m2)+1,min(xe2)-1,max(xe2)+1]);
xlabel('n','FontSize',8); ylabel('x_e(n)','FontSize',8);
title('Even part of x_2(n) = exp(0.1n) [u(n + 5) - u(n -
10)]',...
'FontSize',8);
ntick = [m2(1):2:m2(end)]; set(gca,'XTick',ntick);
subplot(2,1,2); Hs = stem(m2,xo2,'filled');
set(Hs,'markersize',2);
```

```
axis([min(m2)-1,max(m2)+1,min(xo2)-1,max(xo2)+1]);
xlabel('n','FontSize',8); ylabel('x_o(n)','FontSize',8);
title('Odd part of x_2(n) = exp(0.1n) [u(n + 5) - u(n -
10)]',...
'FontSize',8);
ntick = [m2(1) :2:m2(end)]; set(gca,'XTick',ntick);
print -deps2 ../EPSFILES/P0206b;
```

The plots of $x_2(n)$ is shown in Figure 2.24.

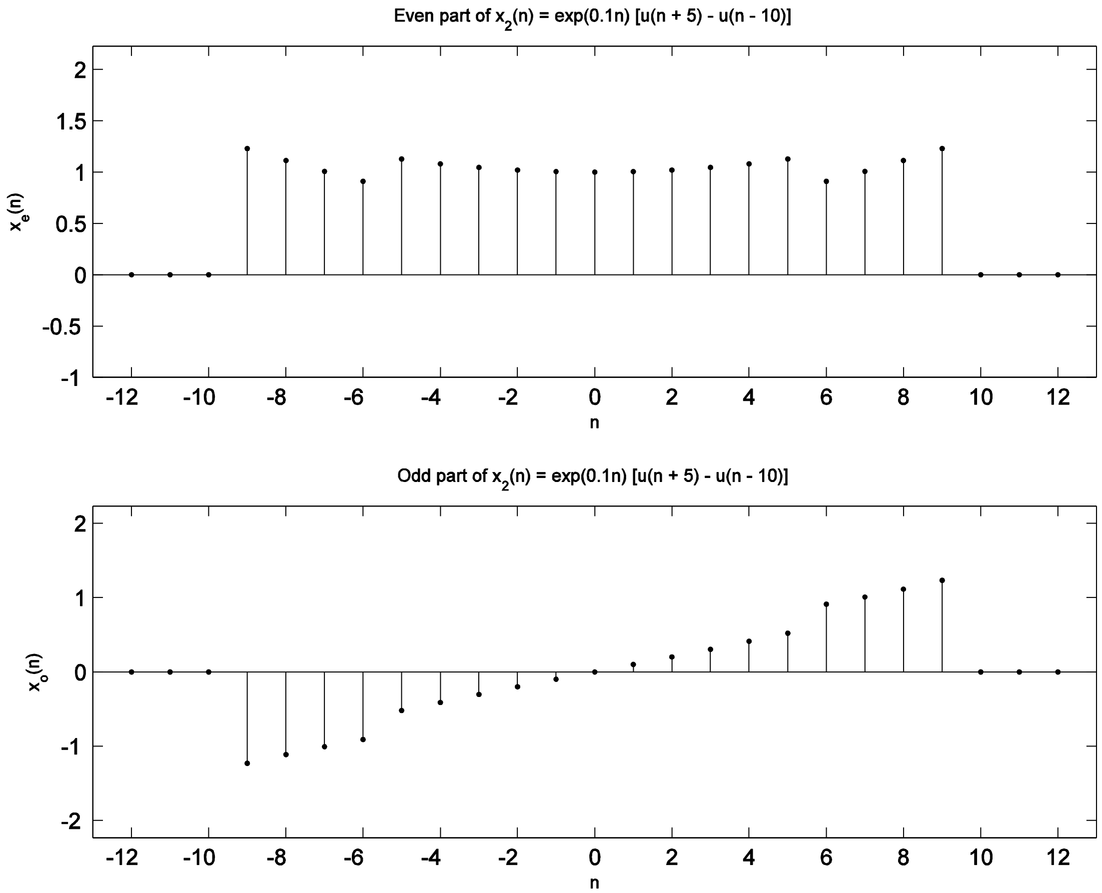


Figure 2.24: Problem P2.6.2 sequence plot

3. $x_3(n) = \cos(0.2\pi n + \pi/4)$, $-20 \le n \le 20$

```
%% P0206c: Even odd decomposition of x2(n) = cos(0.2*pi*n
+ pi/4);
% -20 <= n <= 20;
clc; close all;
n3 = [-20:20]; x3 = cos(0.2*pi*n3 + pi/4);
[xe3,xo3,m3] = evenodd(x3,n3);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0206c');
subplot(2,1,1); Hs = stem(m3,xe3,'filled');
set(Hs,'markersize',2);
```

```
axis([min(m3)-2,max(m3)+2,min(xe3)-1,max(xe3)+1]);
xlabel('n','FontSize',8); ylabel('x_e(n)','FontSize',8);
title('Even part of x_3(n) = cos(0.2 \times \pi \times n
+ \pi/4)',...
'FontSize',8);
ntick = [m3(1):4:m3(end)]; set(gca,'XTick',ntick);
subplot(2,1,2); Hs = stem(m3,xo3,'filled');
set(Hs,'markersize',2);
axis([min(m3)-2,max(m3)+2,min(xo3)-1,max(xo3)+1]);
xlabel('n','FontSize',8); ylabel('x_o(n)','FontSize',8);
title('Odd part of x_3(n) = cos(0.2 \times \pi \times n +
\pi/4)',...
'FontSize',8);
ntick = [m3(1):4 :m3(end)]; set(gca,'XTick',ntick);
print -deps2 ../EPSFILES/P0206c;
```

The plots of $x_3(n)$ is shown in Figure 2.25.

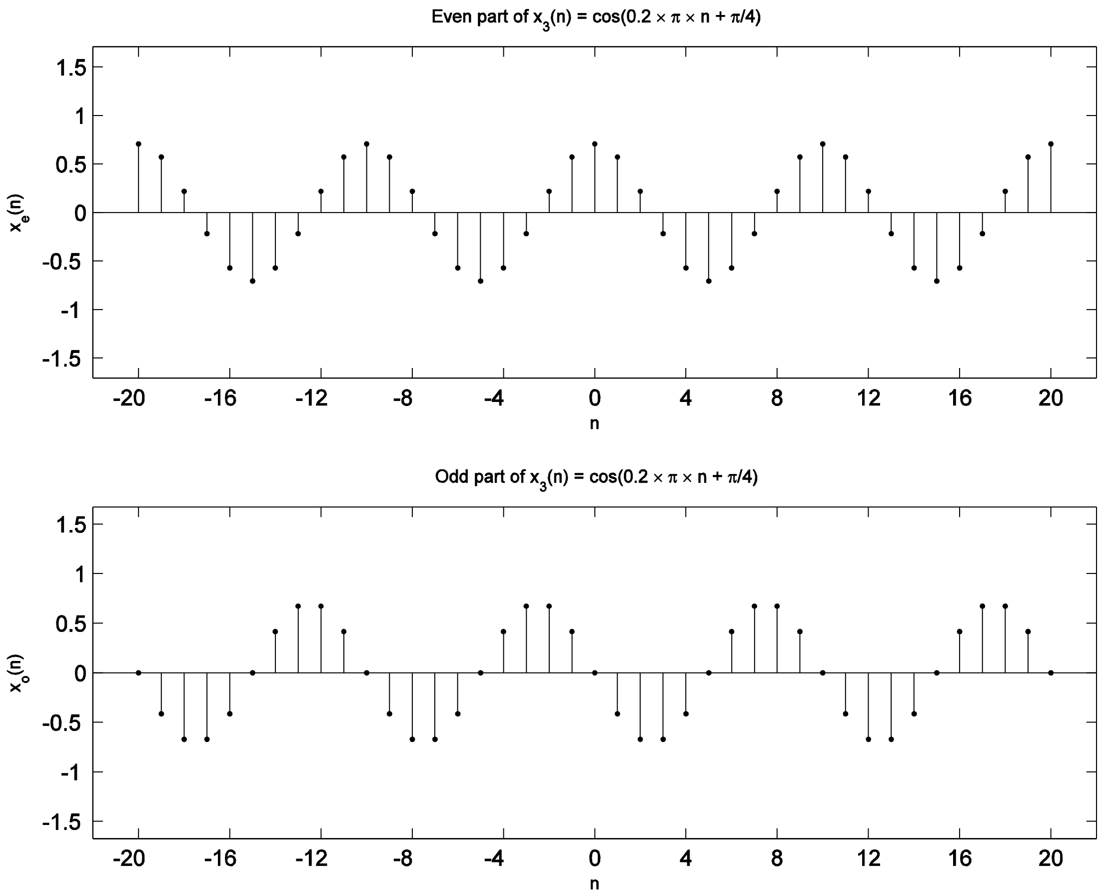


Figure 2.25: Problem P2.6.3 sequence plot

4. $x_4(n) = e^{-0.005n}\sin(0.1\pi n + \pi/3),\ 0 \le n \le 100$

```
%% P0206d: x4(n) = e ^ {-0.05*n}*sin(0.1*pi*n + pi/3), 0
<= n <= 100
```

```
clc; close all;
n4 = [0:100]; x4 = exp(-0.05*n4).*sin(0.1*pi*n4 + pi/3);
[xe4,xo4,m4] = evenodd(x4,n4);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0206d');
subplot(2,1,1); Hs = stem(m4,xe4,'filled');
set(Hs,'markersize',2);
axis([min(m4)-10,max(m4)+10,min(xe4)-1,max(xe4)+1]);
xlabel('n','FontSize',8); ylabel('x_e(n)','FontSize',8);
title(['Even part of x_4(n) = ' ...
'exp(-0.05 \times n) \times sin(0.1 \times \pi \times n +
' ...
'\pi/3)'],'FontSize',8);
ntick = [m4(1):20:m4(end)]; set(gca,'XTick',ntick);
subplot(2,1,2); Hs = stem(m4,xo4,'filled');
set(Hs,'markersize',2);
axis([min(m4)-10,max(m4)+10,min(xo4)-1,max(xo4)+1]);
xlabel('n','FontSize',8); ylabel('x_o(n)','FontSize',8);
title(['Odd part of x_4(n) =' ...
'exp(-0.05 \times n) \times sin(0.1 \times \pi \times n +
' ...
'\pi/3)'],'FontSize',8);
ntick = [m4(1):20 :m4(end)]; set(gca,'XTick',ntick);
print -deps2 ../EPSFILES/P0206d;
```

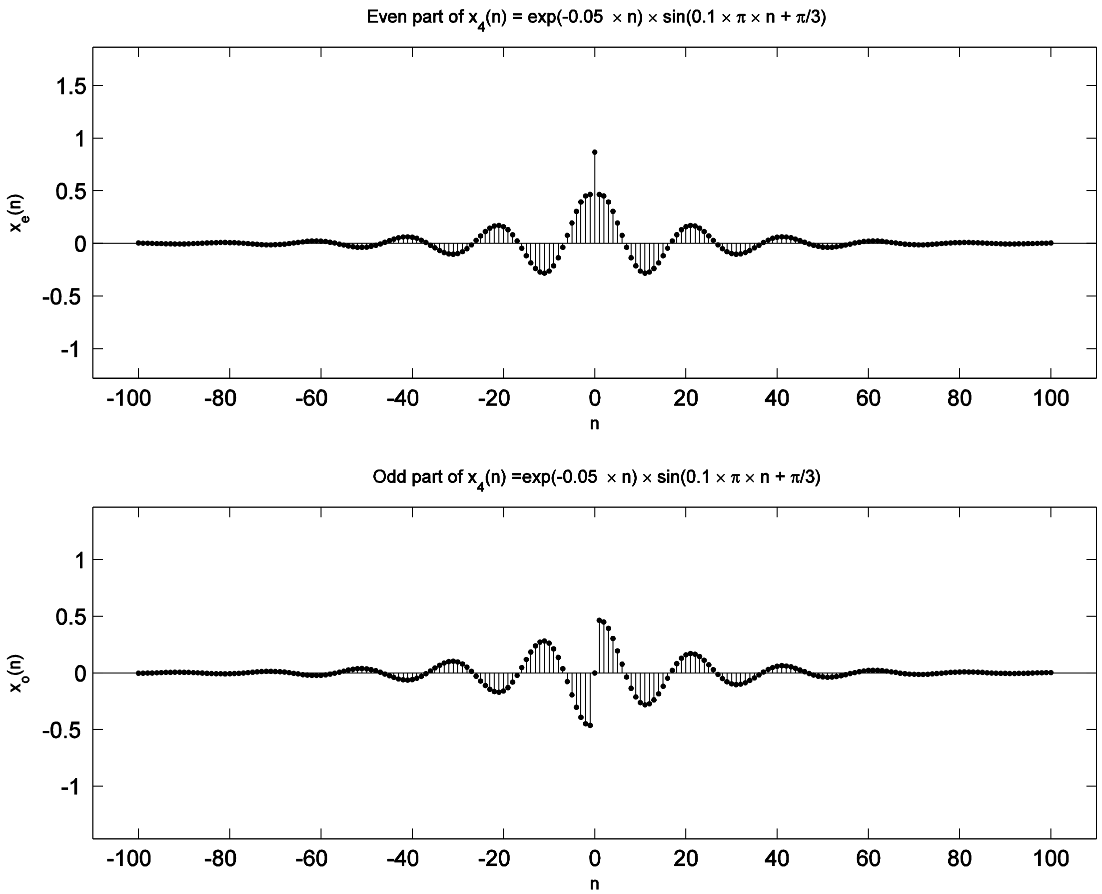


Figure 2.26: Problem P2.6.1 sequence plot

P2.7

A complex-valued sequence $\mathbf{x}_e(n)$ is called *conjugate-symmetric* if $\mathbf{x}_e(n) = \mathbf{x}_e^*(-n)$ and a complex-valued sequence $\mathbf{x}_o(n)$ is called *conjugate-antisymmetric* if $\mathbf{x}_o(n) = -\mathbf{x}_o^*(-n)$. Then, any arbitrary complex-valued sequence $x(n)$ can be decomposed into $x(n) = \mathbf{x}_e(n) + \mathbf{x}_o(n)$ where $\mathbf{x}_e(n)$ and $\mathbf{x}_o(n)$ are given by

$$\mathbf{x}_e(n) = \frac{1}{2}\,[x(n) + \mathbf{x}^*(-n)] \qquad \text{and} \qquad \mathbf{x}_o(n) = \frac{1}{2}[x(n) - \mathbf{x}^*(-n)] \qquad \textbf{(2.27)}$$

respectively.

1. Modify the **evenodd** function discussed in the text so that it accepts an arbitrary sequence and decomposes it into its conjugate-symmetric and conjugate-antisymmetric components by implementing (2.27)
2. Decompose the following sequence:

$$x(n) = 10\exp([-0.1 + j0.2\pi]n), \quad 0 \le n \le 10$$

into its conjugate-symmetric and conjugate-antisymmetric components. Plot their real and imaginary parts to verify the decomposition. (Use the **subplot** function.)

Solutions

1. Modify the **evenodd** function discussed in the text so that it accepts an arbitrary sequence and decomposes it into its conjugate-symmetric and conjugate-antisymmetric components by implementing (2.27).

```
function [xe,xo,m] = evenodd(x,n)
m = -fliplr(n);
m1 = min([m,n]);
m2 = max([m,n]);
m = m1:m2;
nm = n(1)-m(1);
n1 = 1:length(n);
x1 = zeros(1,length(m));
x1(n1+nm) = x;
x = x1;
xe = 0.5*(x + fliplr(conj(x)));
xo = 0.5*(x-fliplr(conj(x)));
```

2. $x(n) = 10 \exp([-0.1 + j0.2\pi]n)$, $0 \leq n \leq 10$

```
% P2.7
% P0207b: Decomposition of x(n) = 10*e ^ {(-0.1 +
j*0.2*pi)*n},
% 0 < = n <= 10
% into its conjugate symmetric and conjugate
antisymmetric parts.
clc; close all;
n = [0:10]; x = 10*(-0.1+1i*0.2*pi)*n; [xe,xo,neo] =
evenodd(x,n);
% function [xe,xo,m] = evenodd(x,n) is modified to be
suitable for arbitrary sequence.
Re_xe = real(xe); Im_xe = imag(xe); Re_xo = real(xo);
Im_xo = imag(xo);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0207b');
subplot(2,2,1); Hs = stem(neo,Re_xe);
set(Hs,'markersize',2);
ylabel('Re[x_e(n)]','FontSize',8);
xlabel('n','FontSize',8);
axis([min(neo)-1,max(neo)+1,min(Re_xe)-2,max(Re_xe)+2]);
ytick = [min(Re_xe)-2:2:max(Re_xe)+2];
set(gca,'YTick',ytick);
title(['Real part of' char(10) 'even sequence
```

```
x_e(n)'],'FontSize',8);
subplot(2,2,3); Hs = stem(neo,Im_xe);
set(Hs,'markersize',2);
ylabel('Im[x_e(n)]','FontSize',8);
xlabel('n','FontSize',8);
axis([min(neo)-1,max(neo)+1,-40,40]);
ytick = [-40:20:40]; set(gca,'YTick',ytick);
title(['Imaginary part of' char(10) 'even sequence
x_e(n)'],'FontSize',8);
subplot(2,2,2); Hs = stem(neo,Re_xo);
set(Hs,'markersize',2);
ylabel('Re[x_o(n)]','FontSize',8);
xlabel('n','FontSize',8);
axis([min(neo)-1,max(neo)+1,min(Re_xo)-1,max(Re_xo)+1]);
ytick = [-6:2:6]; set(gca,'YTick',ytick);
title(['Real part of' char(10) 'odd sequence
x_o(n)'],'FontSize',8);
subplot(2,2,4); Hs = stem(neo,Im_xo);
set(Hs,'markersize',2);
ylabel('Im[x_o(n)]','FontSize',8);
xlabel('n','FontSize',8);
axis([min(neo)-1,max(neo)+1,min(Im_xo)-
10,max(Im_xo)+10]);
ytick = [-10:10:40]; set(gca,'YTick',ytick);
title(['Imaginary part of' char(10) 'odd sequence
x_o(n)'],'FontSize',8);
print -deps2 ../EPSFILES/P0207b;
```

The plots of $x(n)$ are shown in Figure 2.27.

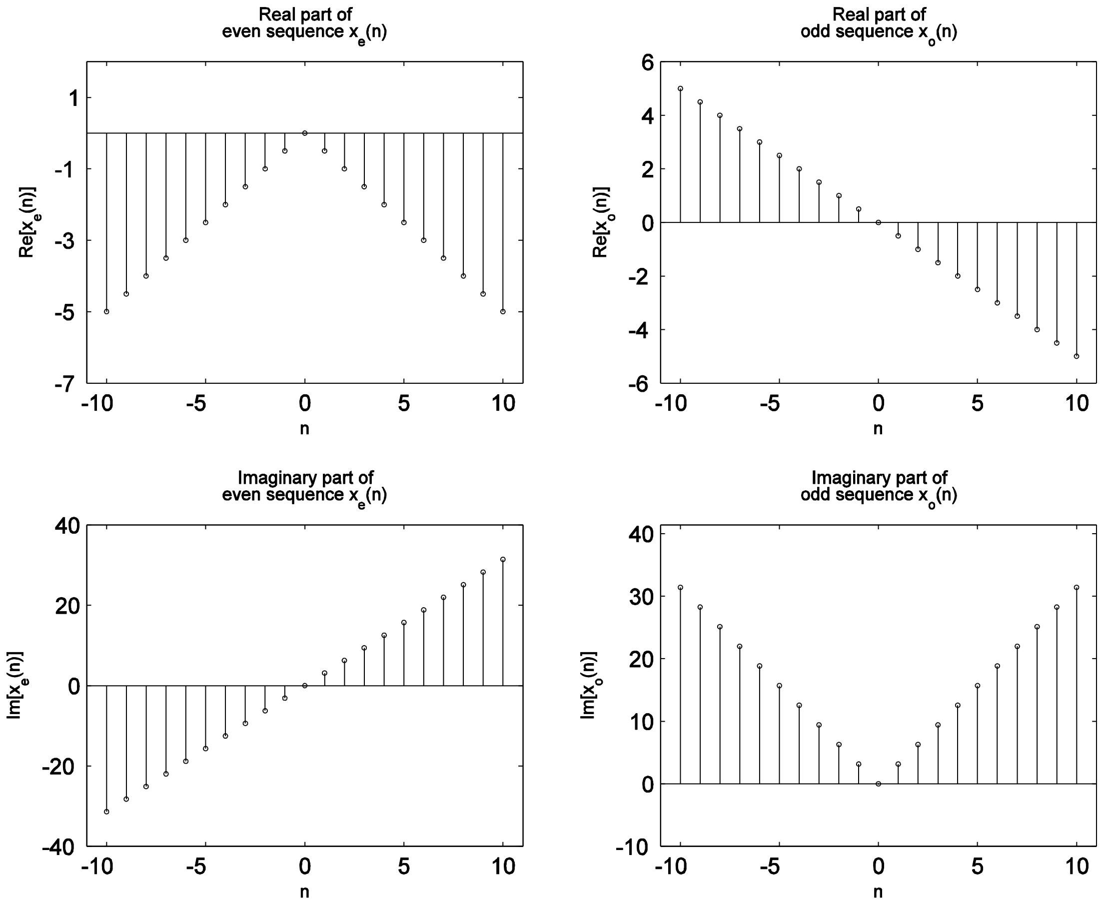


Figure 2.27: Problem P2.7.2 sequence plot

P2.8

The operation of *signal dilation* (or *decimation* or *down-sampling*) is defined by

$$y(n) = x(nM)$$

in which the sequence $x(n)$ is down-sampled by an integer factor M. For example, if

$$x(n) = \{\ldots, -2, 4, \underset{\uparrow}{3}, -6, 5, -1, 8, \ldots\}$$

then the down-sampled sequences by a factor 2 are given by

$$y(n) = \{\ldots, -2, \underset{\uparrow}{3}, 5, 8, \ldots\}$$

1. Develop a MATLAB function **dnsample** that has the form

```
function [y,m]=dnsample(x,n,M)
% Downsample sequence x(n) by a factor M to obtain y(m)
```

to implement the above operation. Use the indexing mechanism of MATLAB with careful attention to the origin of the time axis $n = 0$.

2. Generate $x(n) = \sin(0.125\pi n)$, $-50 \le n \le 50$. Decimate $x(n)$ by a factor of 4 to generate $y(n)$.

Plot both $x(n)$ and $y(n)$ using subplot and comment on the results.

3. Repeat the above using $x(n) = \sin(0.5\pi n)$, $-50 \le n \le 50$. Qualitatively discuss the effect of down-sampling on signals.

Solutions

1. Matlab function:

```
function [y,m]=dnsample(x,n,M)
% Downsample sequence x(n) by a factor M to obtain y(m)

%x is a sequence over indices specified by vector n,M is
the downsampling factor.

param=n/M;
%generates the parameter vector.This vector will decide
which input samples will be present in the output.

samp=fix(param)==param;
%only those output vectors corresponding to indices where
samp==1 will be present in the output.

y=x(samp==1);
%generates the output sequence

m=n(samp==1)/M;
%generates the indices of the output sequence

End
```

2. $x_1(n) = \sin(0.125\pi n)$, $-50 \le n \le 50$. Decimation by a factor of 4.

```
% P2.8
%% P0208b: x1(n) = sin(0.125*pi*n),-50 <= n <= 50
% Decimate x(n) by a factor of 4 to obtain y(n)
clc; close all;
n1 = [-50:50]; x1 = sin(0.125*pi*n1); [y1,m1] =
dnsample(x1,n1,4);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0208b');
subplot(2,1,1); Hs = stem(n1,x1); set(Hs,'markersize',2);
xlabel('n','FontSize',8); ylabel('x(n)','FontSize',8);
title('Original sequence x_1(n)','FontSize',8);
axis([min(n1)-5,max(n1)+5,min(x1)-0.5,max(x1)+0.5]);
ytick = [-1.5:0.5:1.5]; ntick = [n1(1):10:n1(end)];
```

```
set(gca,'XTick',ntick); set(gca,'YTick',ytick);
subplot(2,1,2); Hs = stem(m1,y1); set(Hs,'markersize',2);
xlabel('n','FontSize',8); ylabel('y(n) =
x(4n)','FontSize',8);
title('y_1(n) = Original sequence x_1(n) decimated by a
factor of 4',...
'FontSize',8);
axis([min(m1)-2,max(m1)+2,min(y1)-0.5,max(y1)+0.5]);
ytick = [-1.5:0.5:1.5]; ntick = [m1(1):2:m1(end)];
set(gca,'XTick',ntick); set(gca,'YTick',ytick);
print -deps2 ../EPSFILES/P0208b;
```

The plots of $x_1(n)$ and $y_1(n)$ are shown in Figure 2.28. Observe that the original signal $x_1(n)$ can be recovered.

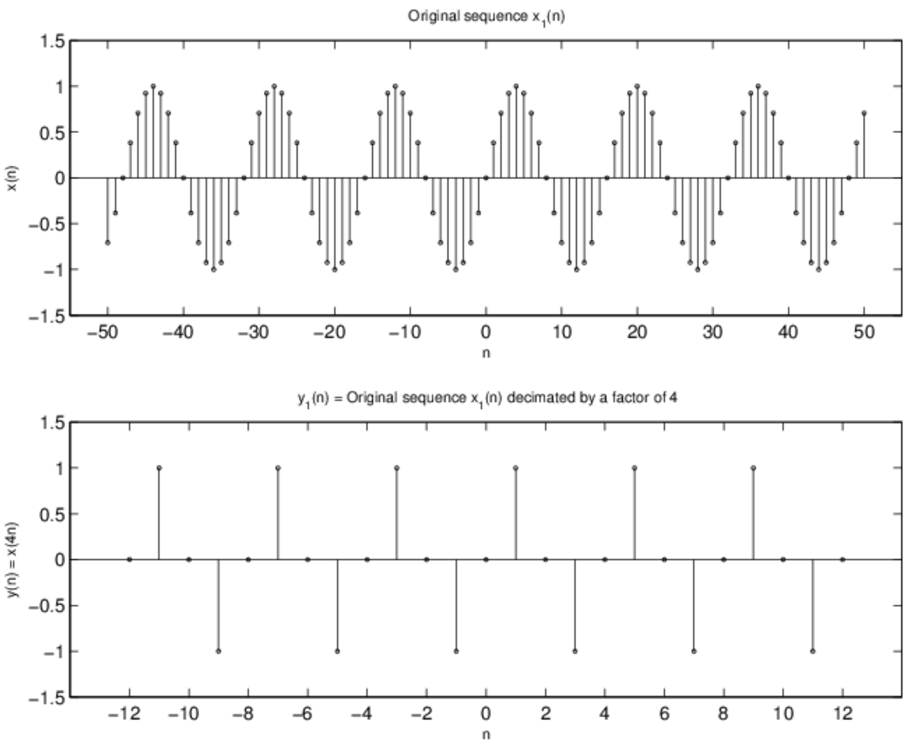


Figure 2.28: Problem P2.8.2 sequence plot

3. $x(n) = \sin(0.5\pi n)$, $-50 \le n \le 50$. Decimation by a factor of 4.

```
%% P0208c: x2(n) = sin(0.5*pi*n),-50 <= n <= 50
% Decimate x2(n) by a factor of 4 to obtain y2(n)
clc; close all;
n2 = [-50:50]; x2 = sin(0.5*pi*n2); [y2,m2] =
dnsample(x2,n2,4);
Hf_1 = figure;
```

```
set(Hf_1,'NumberTitle','off','Name','P0208c');
subplot(2,1,1); Hs = stem(n2,x2); set(Hs,'markersize',2);
xlabel('n','FontSize',8); ylabel('x(n)','FontSize',8);
axis([min(n2)-5,max(n2)+5,min(x2)-0.5,max(x2)+0.5]);
title('Original sequence x_2(n)','FontSize',8);
ytick = [-1.5:0.5:1.5]; ntick = [n2(1):10:n2(end)];
set(gca,'XTick',ntick); set(gca,'YTick',ytick);
subplot(2,1,2); Hs = stem(m2,y2); set(Hs,'markersize',2);
xlabel('n','FontSize',8); ylabel('y(n) =
x(4n)','FontSize',8);
axis([min(m2)-1,max(m2)+1,min(y2)-1,max(y2)+1]);
title('y_2(n) = Original sequence x_2(n) decimated by a
factor of 4',...
'FontSize',8);
ntick = [m2(1):2:m2(end)]; set(gca,'XTick',ntick);
print -deps2 ../EPSFILES/P0208c;
```

The plots of $x_2(n)$ and $y_2(n)$ are shown in Figure 2.29. Observe that the downsampled signal is a signal with zero frequency. Thus the original signal $x_2(n)$ is lost.

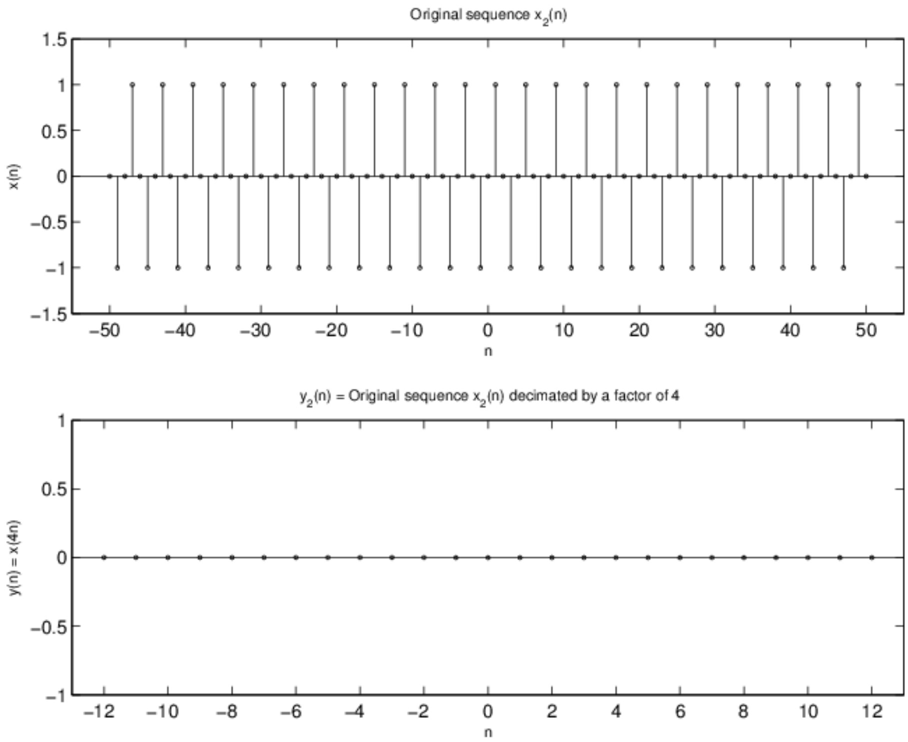


Figure 2.29: Problem P2.8.3 sequence plot

P2.9

Using the **conv_m** function, determine the autocorrelation sequence $\mathbf{r}_{xx}()$ and the crosscorrelation sequence $\mathbf{r}_{xy}()$ for the following sequences.

$$x(n) = (0.9)^n, \; 0 \le n \le 20; \quad y(n) = (0.8)^{-n}, \; -20 \le n \le 0$$

Describe your observations of these results.

Solutions

```
% P2.9
%% P0209a: autocorrelation of sequence x(n) = 0.9 ^ n, 0
<= n <= 20
% using the conv_m function
clc; close all;
nx = [0:20]; x = 0.9 .^ nx; [xf,nxf] = sigfold(x,nx);
[rxx,nrxx] = conv_m(x,nx,xf,nxf);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0209a');
Hs = stem(nrxx,rxx); set(Hs,'markersize',2);
xlabel('n','FontSize',8);
ylabel('r_x_x(n)','FontSize',8);
title('Autocorrelation of x(n)','FontSize',8);
axis([min(nrxx)-1,max(nrxx)+1,min(rxx),max(rxx)+1]);
ntick = [nrxx(1):4:nrxx(end)]; set(gca,'XTick',ntick);
print -deps2 ../EPSFILES/P0209a;
```

The plot of the autocorrelation is shown in Figure 2.30.

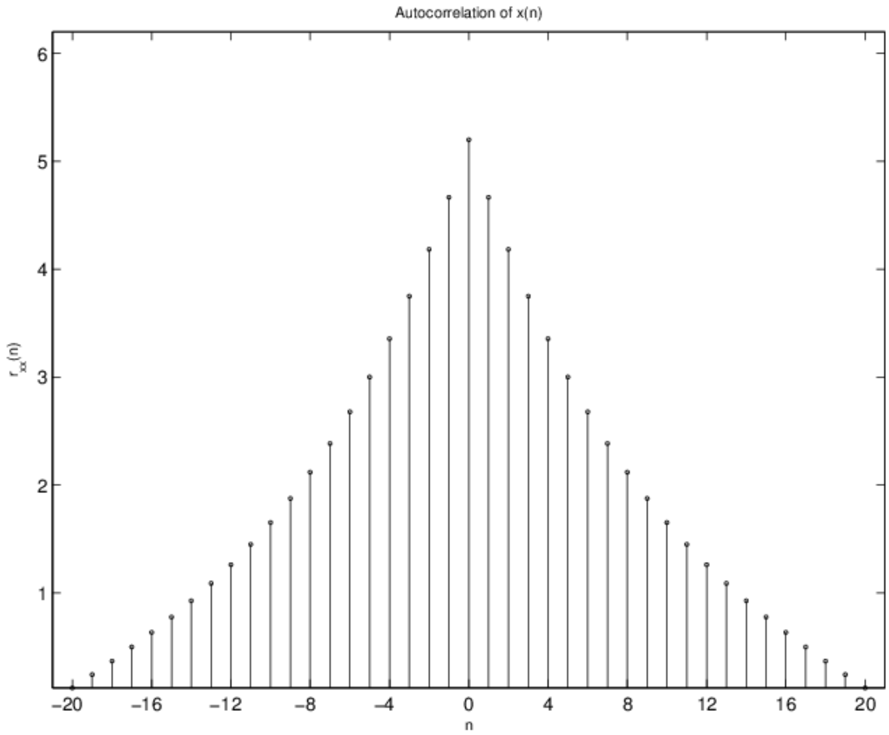


Figure 2.30: Problem P2.9 autocorrelation plot

```
%% P0209b: crosscorrelation of sequence x(n) = 0.9 ^ n, 0
<= n <= 20
% with sequence y = 0.8.^n, -20 <=n <= 0 using the conv_m
function
clc; close all;
nx = [0:20]; x = 0.9 .^ nx; ny = [-20:0]; y = 0.8 .^ ny;
[yf,nyf] = sigfold(y,ny); [rxy,nrxy] =
conv_m(x,nx,yf,nyf);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0209b');
Hs = stem(nrxy,rxy); set(Hs,'markersize',2);
xlabel('n','FontSize',8);
ylabel('r_x_y(n)','FontSize',8);
title('Crosscorrelation of x(n) and y(n)','FontSize',8);
axis([min(nrxy)-1,max(nrxy)+1,min(rxy)-1,max(rxy)+20]);
ytick = [0:50:300 320]; ntick = [nrxy(1):2:nrxy(end)];
set(gca,'XTick',ntick); set(gca,'YTick',ytick);
print -deps2 ../EPSFILES/P0209b;
```

The plot of the crosscorrelation is shown in Figure 2.31.

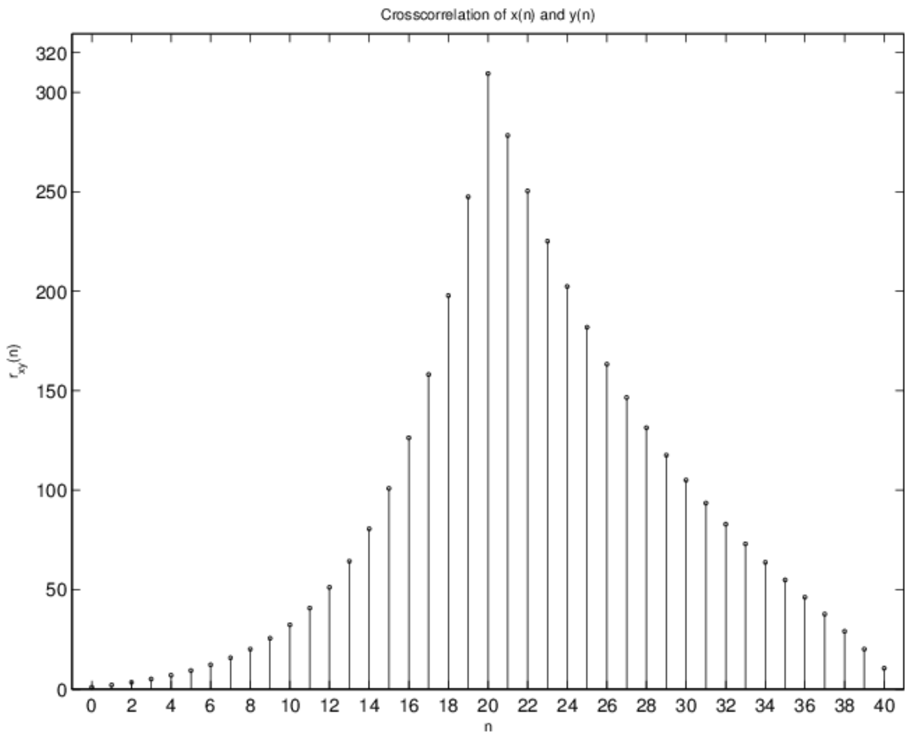


Figure 2.31: Problem P2.9 crosscorrelation plot

P2.10

In a certain concert hall, echoes of the original audio signal $x(n)$ are generated due to the reflections at the walls and ceiling. The audio signal experienced by the listener $y(n)$ is a combination of $x(n)$ and its echoes. Let $y(n) = x(n)+\alpha x(n - k)$ where k is the amount of delay in samples and α is its relative strength. We want to estimate the delay using the correlation analysis.

1. Determine analytically the autocorrelation $\mathrm{r}_{yy}(\ell)$ in terms of the autocorrelation $\mathrm{r}_{xx}(\ell)$.
2. Let $x(n) = \cos(0.2\pi n) + 0.5 \cos(0.6\pi n)$, $\alpha = 0.1$, and $k = 50$. Generate 200 samples of $y(n)$ and determine its autocorrelation. Can you obtain α and k by observing $\mathrm{r}_{yy}(\ell)$?

Solutions

```
%% P0210a: To prove the fomula below:
% r_y_y(n) = conv_m(y(n),y(-n)) =
% conv_m(x(n),x(-n))+a*conv_m(x(n),x(-n-k)) +
a*conv_m(x(n-k),x(-n)+a*a*conv_m(x(n-k),x(-n-k))
[xf,nxf] = sigfold(x,nx);
[rxx,nrxx] = conv_m(x,nx,xf,nxf);
[x2,nx2] = sigshift(xf,nxf,-k);
[rxx2,nrxx2] = conv_m(x,nx,x2,nx2);
```

```
[ryy2,nryy2] = sigadd(rxx,nrxx,a*rxx2,nrxx2);
[rx1xf,nrx1xf] = conv_m(x1,nx1,xf,nxf);
[ryy2,nryy2]  = sigadd(ryy2,nryy2,a*rx1xf,nrx1xf);
[x1f,nx1f] = sigfold(x1,nx1);
[rx1x1,nrx1x1] = conv_m(x1,nx1,x1f,nx1f);
[ryy2,nryy2]  = sigadd(ryy2,nryy2,a*a*rx1x1,nrx1x1);
error_y = ryy2-ryy;
error_ny = nryy2-nryy;
max(abs(error_y));

% P2.10
% P0210b: autocorrelation of sequence y(n) = x(n) +
a*x(n-k)
% x(n) = cos(0.2*pi*n) + 0.5cos(0.6*pi*n)
% Generate 200 samples of y(n) and use the conv_m
function
nx = 1:150;x = cos(0.2*pi*nx)+0.5*cos(0.6*pi*nx);a =
0.1;k = 50;
[x1,nx1]=sigshift(x,nx,k);[y,ny] = sigadd(x,nx,a*x1,nx1);
[yf,nyf]=sigfold(y,ny);[ryy,nryy] = conv_m(y,ny,yf,nyf);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0210b');
Hs = stem(nryy,ryy); set(Hs,'markersize',2);
xlabel('n','FontSize',8);
ylabel('r_y_y(n)','FontSize',8);
title('Autocorrelation of y(n)','FontSize',8);
axis([min(nryy)-1,max(nryy)+1,min(ryy)-10,max(ryy)+10]);
ytick = [-110 -90:30:90 110]; ntick = [nryy(1)-
1:10:nryy(end)+1];
set(gca,'XTick',ntick); set(gca,'YTick',ytick);
print -deps2 P0210b;
```

The plot of the autocorrelation is shown in Figure 2.32.

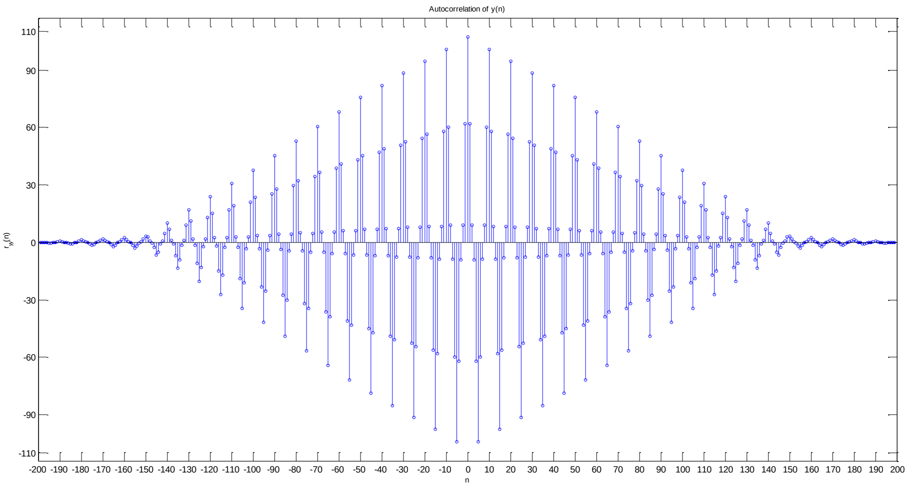


Figure 2.32: Problem P2.10 autocorrelation plot

P2.11

Consider the following discrete-time systems:

$T_1[x(n)] = x(n)u(n)$ $\qquad$ $T_2[x(n)] = x(n) + n\,x(n+1)$

$T_3[x(n)] = x(n) + \frac{1}{2}x(n-2) - \frac{1}{3}x(n-3)x(2n)$ $\qquad$ $T_4[x(n)] = \sum_{k=-\infty}^{n+5} 2x(k)$

$T_5[x(n)] = x(2n)$ $\qquad$ $T_6[x(n)] = \text{round}[x(n)]$

where round[·] denotes rounding to the nearest integer.

1. Use (2.10) to determine analytically whether these systems are linear.
2. Let $x_1(n)$ be a uniformly distributed random sequence between [0, 1] over $0 \le n \le 100$, and let $x_2(n)$ be a Gaussian random sequence with mean 0 and variance 10 over $0 \le n \le 100$. Using these sequences, verify the linearity of these systems. Choose any values for constants a_1 and a_2 in (2.10). You should use several realizations of the above sequences to arrive at your answers.

Solutions

System-1: $T_1[x(n)] = x(n)u(n)$

1. Analytic determination of linearity:

$T_1[a_1x_1(n) + a_2x_2(n)] = \{a_1x_1(n) + a_2x_2(n)\}u(n) = a_1x_1(n)u(n) + a_2x_2(n)u(n) = a_1T_1[x_1(n)] + a_2T_1[x_2(n)]$

Hence the system $T_1[x(n)]$ is **linear**.

2. Matlab script:

```
% P2.11
%% P0211a: To prove that the system T1[x(n)] = x(n)u(n)
is linear
```

```
clear; clc; close all;
n = 0:100; x1 = rand(1,length(n));
x2 = sqrt(10)*randn(1,length(n)); u = stepseq(0,0,100);
y1 = x1.*u; y2 = x2.*u; y = (x1 + x2).*u;
diff = sum(abs(y - (y1 + y2)));
if (diff < 1e-5)
disp(' *** System-1 is Linear *** ');
else
disp(' *** System-1 is NonLinear ***');
end
```

Matlab verification:
>> *** System-1 is Linear ***

System-2: $T_2[x(n)] = x(n) + n\,x(n+1)$

1. Analytic determination of linearity:

$T_2[a_1x_1(n) + a_2x_2(n)] = \{a_1x_1(n) + a_2x_2(n)\} + n\,\{a_1x_1(n+1) + a_2x_2(n+1)\} = a_1\{x_1(n) + n\,x_1(n+1)\} + a_2\{x_2(n) + n\,x_2(n+1)\} = a_1T_2[x_1(n)] + a_2T_2[x_2(n)]$

Hence the system is $T_2[x(n)]$ **linear**.

2. Matlab script:

```
%% P0211b: To prove that the system T2[x(n)] = x(n) +
n*x(n+1) is linear
clear; clc; close all;
n = 0:100; x1 = rand(1,length(n)); x2 =
sqrt(10)*randn(1,length(n));
z = n; [x11,nx11] = sigshift(x1,n,-1);
[x111,nx111] = sigmult(z,n,x11,nx11); [y1,ny1] =
sigadd(x1,n,x111,nx111);
[x21,nx21] = sigshift(x2,n,-1); [x211,nx211] =
sigmult(z,n,x21,nx21);
[y2,ny2] = sigadd(x2,n,x211,nx211);
xs = x1 + x2; [xs1,nxs1] = sigshift(xs,n,-1);
[xs11,nxs11] = sigmult(z,n,xs1,nxs1); [y,ny] =
sigadd(xs,n,xs11,nxs11);
diff = sum(abs(y - (y1 + y2)));
if (diff < 1e-5)
disp(' *** System-2 is Linear *** ');
else
disp(' *** System-2 is NonLinear *** ');
end
```

Matlab verification:
>> *** System-2 is Linear ***

System-3: $T_3[x(n)] = x(n) + \frac{1}{2}x(n-2) - \frac{1}{3}x(n-3)x(2n)$

1. Analytic determination of linearity:

$T_3[a_1x_1(n) + a_2x_2(n)] = a_1x_1(n) + a_2x_2(n) + \frac{1}{2}\{a_1x_1(n - 2) + a_2x_2(n - 2)\} - \frac{1}{3}\{a_1x_1(n - 3) + a_2x_2(n - 3)\}\{a_1x_1(2n) + a_2x_2(2n)\} = a_1x_1(n) + \frac{1}{2}x_1(n - 2) - \frac{1}{3}a_1x_1(n - 3)x_1(2n) + a_2x_2(n) + \frac{1}{2}x_2(n - 2) - \frac{1}{3}a_2x_2(n - 3)x_2(2n) + \frac{1}{3}\{a_1x_1(n - 3)a_2x_2(2n) + a_2x_2(n - 3)a_1x_1(2n)\}$

which clearly is not equal to $a_1T_3[x_1(n)]+a_2T_3[x_2(n)]$. The product term in the input-output equation makes the system $T_3[x(n)]$ **nonlinear**.

2. Matlab script:

```
%% P0211c: To prove that the system T3[x(n)] = x(n) +
1/2*x(n - 2)
% - 1/3*x(n - 3)*x(2n)
% is linear
clear; clc; close all;
n = [0:100]; x1 = rand(1,length(n)); x2 =
sqrt(10)*randn(1,length(n));
[x11,nx11] = sigshift(x1,n,2); x11 = 1/2*x11; [x12,nx12]
= sigshift(x1,n,3);
x12 = -1/3*x12; [x13,nx13] = dnsample(x1,n,2);
[x14,nx14] = sigmult(x12,nx12,x13,nx13);
[x15,nx15] = sigadd(x11,nx11,x14,nx14);
[y1,ny1] = sigadd(x1,n,x15,nx15); [x21,nx21] =
sigshift(x2,n,2);
x21 = 1/2*x21; [x22,nx22] = sigshift(x2,n,3);
x22 = -1/3*x22; [x23,nx23] = dnsample(x2,n,2);
[x24,nx24] = sigmult(x22,nx22,x23,nx23);
[x25,nx25] = sigadd(x21,nx21,x24,nx24); [y2,ny2] =
sigadd(x2,n,x25,nx25);
xs = x1 + x2; [xs1,nxs1] = sigshift(xs,n,2);
xs1 = 1/2*xs1; [xs2,nxs2] = sigshift(xs,n,3); xs2 = -
1/3*xs2;
[xs3,nxs3] = dnsample(xs,n,2); [xs4,nxs4] =
sigmult(xs2,nxs2,xs3,nxs3);
[xs5,nxs5] = sigadd(xs1,nxs1,xs4,nxs4);
[y,ny] = sigadd(xs,n,xs5,nxs5); diff = sum(abs(y - (y1 +
y2)));
if (diff < 1e-5)
disp(' *** System-3 is Linear *** ');
else
disp(' *** System-3 is NonLinear *** ');
end
```

Matlab verification:

>> *** System-3 is NonLinear ***

System-4: $T_4[x(n)] = \sum_{k-\infty}^{n+5} 2x(k)$

1. Analytic determination of linearity:

$T_4[a_1x_1(n) + a_2x_2(n)] = \sum_{k=-\infty}^{n+5} 2\{a_1x_1(k) + a_2x_2(k)\} = a_1 \sum_{k=-\infty}^{n+5} 2x_1(k) + a_2 \sum_{k=-\infty}^{n+5} 2x_2(k) = a_1T_4[x_1(n)] + a_2T_4[x_2(n)]$

Hence the system $T_4[x(n)]$ is **linear**.

2. Matlab script:

```
%% P0211d: To prove that the system T4[x(n)] = sum_{k=-
infinity}^{n+5}2*x(k)
% is linear
clear; clc; close all;
n = [0:100]; x1 = rand(1,length(n)); x2 =
sqrt(10)*randn(1,length(n));
y1 = cumsum(x1); ny1 = n - 5; y2 = cumsum(x2); ny2 = n -
5; xs = x1 + x2;
y = cumsum(xs); ny = n - 5; diff = sum(abs(y - (y1 +
y2)));
if (diff < 1e-5)
disp('*** System-4 is Linear *** ');
else
disp(' *** System-4 is NonLinear *** ');
end
```

Matlab verification:

>> *** System-4 is Linear ***

System-5: $T_5[x(n)] = x(2n)$

1. Analytic determination of linearity:

$T_5[a_1x_1(n) + a_2x_2(n)] = a_1x_1(2n) + a_2x_2(2n) = a_1T_5[x_1(n)] + a_2T_5[x_2(n)]$

Hence the system $T_5[x(n)]$ is **linear**.

2. Matlab script:

```
%% P0211e: To prove that the system T5[x(n)] = x(2n) is
linear
clear; clc; close all;
n = 0:100; x1 = rand(1,length(n)); x2 =
sqrt(10)*randn(1,length(n));
[y1,ny1] = dnsample(x1,n,2); [y2,ny2] = dnsample(x2,n,2);
xs = x1 + x2;
[y,ny] = dnsample(xs,n,2); diff = sum(abs(y - (y1 +
y2)));
if (diff < 1e-5)
disp('*** System-5 is Linear *** ');
else
disp(' *** System-5 is NonLinear *** ');
end
```

Matlab verification:
>> *** System-5 is Linear ***

System-6: $T_6[x(n)] = \text{round}[x(n)]$
1. Analytic determination of linearity:
$T_6[a_1x_1(n) + a_2x_2(n)] = \text{round}[a_1x_1(n) + a_2x_2(n)] \neq a_1\ \text{round}[x_1(n)] + a_2\ \text{round}[x_2(n)]$
Hence the system $T_6[x(n)]$ is **nonlinear**.
2. Matlab script:
```
%% P0211f: To prove that the system T6[x(n)] =
round(x(n)) is linear
clear; clc; close all;
n = 0:100; x1 = rand(1,length(n)); x2 =
sqrt(10)*randn(1,length(n));
y1 = round(x1); y2 = round(x2); xs = x1 + x2;
y = round(xs); diff = sum(abs(y - (y1 + y2)));
if (diff < 1e-5)
disp(' *** System-6 is Linear *** ');
else
disp(' *** System-6 is NonLinear *** ');
end
```
Matlab verification:
>> *** System-6 is NonLinear ***

P2.12

Consider the discrete-time systems given in Problem P2.11.
1. Use (2.12) to determine analytically whether these systems are time-invariant.
2. Let $x(n)$ be a Gaussian random sequence with mean 0 and variance 10 over $0 \le n \le 100$. Using this sequence, verify the time invariance of the above systems. Choose any values for sample shift k in (2.12). You should use several realizations of the above sequence to arrive at your answers.

Solutions

System-1: $\mathbf{T_1}[x(n)] \triangleq y(n) = x(n)u(n)$
1. Analytic determination of time-invariance:
$\mathbf{T_1}[x(n-k)] = x(n-k)u(n) \neq x(n-k)u(n-k) = y(n-k)$
Hence the system $\mathbf{T_1}[x(n)]$ is **time-varying**.
2. Matlab script:
```
% P2.12
%% P0212a: To determine whether T1[x(n)] = x(n)u(n) is
time invariant
```

```
clear; clc; close all;
n = 0:100; x = sqrt(10)*randn(1,length(n)); u =
stepseq(0,0,100);
y = x.*u; [y1,ny1] = sigshift(y,n,1); [x1,nx1] =
sigshift(x,n,1);
[y2,ny2] = sigmult(x1,nx1,u,n); [diff,ndiff] =
sigadd(y1,ny1,-y2,ny2);
diff = sum(abs(diff));
if (diff < 1e-5)
disp(' *** System-1 is Time-Invariant *** ');
else
disp(' *** System-1 is Time-Varying *** ');
end
```

Matlab verification:

\>> *** System-1 is Time-Varying ***

System-2: $T2[x(n)]$, $y(n) = x(n) + n\,x(n + 1)$

1. Analytic determination of time-invariance:

$T2[x(n - k)] = x(n - k) + n\,x(n - k + 1)$ 6= $x(n - k) + (n - k)\,x(n - k + 1) = y(n - k)$

Hence the system is $T2[x(n)]$ **time-varying**.

2. Matlab script:

```
%% P0212b: To determine whether the system T2[x(n)] =
x(n) + n*x(n + 1) is
% time-invariant
clear; clc; close all;
n = 0:100; x = sqrt(10)*randn(1,length(n));
z = n; [x1,nx1] = sigshift(x,n,-1);
[x11,nx11] = sigmult(z,n,x1,nx1); [y,ny] =
sigadd(x,n,x11,nx11);
[y1,ny1] = sigshift(y,ny ,1); [xs,nxs] = sigshift(x,n,1);
[xs1,nxs1] = sigshift(xs,nxs,-1); [xs11,nxs11] =
sigmult(z,n,xs1,nxs1);
[y2,ny2] = sigadd(xs,nxs,xs11,nxs11); [diff,ndiff] =
sigadd(y1,ny1,-y2,ny2);
diff = sum(abs(diff));
if (diff < 1e-5)
disp(' *** System-2 is Time-Invariant *** ');
else
disp(' *** System-2 is Time-Varying *** ');
end
```

Matlab verification:

\>> *** System-2 is Time-Varying ***

System-3: $T_3[x(n)] \triangleq y(n) = x(n) + \frac{1}{2}x(n - 2) - \frac{1}{3}x(n - 3)x(2n)$

1. Analytic determination of time-invariance:

$T_3[x(n - k)] = x(n - k) + \frac{1}{2}x(n - k - 2) - \frac{1}{3}x(n - k - 3)x(2n - k) \neq x(n - k) + \frac{1}{2}x(n - k - 2) - \frac{1}{3}x(n - k - 3)x(2n - 2k) = y(n - k)$

Hence the system is $T_3[x(n)]$ **time-varying**.

2. Matlab script:

```
%% P0212c: To find whether the system T3[x(n)] = x(n) +
1/2*x(n - 2)
% - 1/3*x(n - 3)*x(2n)
% is time invariant
clear; clc; close all;
n = 0:100; x = sqrt(10)*randn(1,length(n)); [x1,nx1] =
sigshift(x,n,2);
x1 = 1/2*x1; [x2,nx2] = sigshift(x,n,3); x2 = -1/3*x2;
[x3,nx3] = dnsample(x,n,2); [x4,nx4] =
sigmult(x2,nx2,x3,nx3);
[x5,nx5] = sigadd(x1,nx1,x4,nx4); [y,ny] =
sigadd(x,n,x5,nx5);
[y1,ny1] = sigshift(y,ny,1); [xs,nxs] = sigshift(x,n,1);
[xs1,nxs1] = sigshift(xs,nxs,2); xs1 = 1/2*xs1;
[xs2,nxs2] = sigshift(xs,nxs,3); xs2 = -1/3*xs2;
[xs3,nxs3] = dnsample(xs,nxs,2); [xs4,nxs4] =
sigmult(xs2,nxs2,xs3,nxs3);
[xs5,nxs5] = sigadd(xs1,nxs1,xs4,nxs4); [y2,ny2] =
sigadd(xs,nxs,xs5,nxs5);
[diff,ndiff] = sigadd(y1,ny1,-y2,ny2); diff =
sum(abs(diff));
if (diff < 1e-5)
disp(' *** System-3 is Time-Invariant *** ');
else
disp(' *** System-3 is Time-Varying *** ');
end
```

Matlab verification:

>> *** System-3 is Time-Varying ***

System-4: $T_4[x(n)] \triangleq y(n) = \sum_{k=-\infty}^{n+5} 2x(k)$

1. Analytic determination of time-invariance:

$T_4[x(n - \ell)] = \sum_{k=-\infty}^{n+5} 2x(k - \ell) = \sum_{k=-\infty}^{n-l+5} 2x(k) = y(n - \ell)$

Hence the system $T_4[x(n)]$ is **time-invariant**.

2. Matlab script:

```
%% P0212d: To find whether the system T4[x(n)]=sum_{k=-
infinity}^{n+5}2*x(k)
% is time-invariant
clear; clc; close all;
n = 0:100; x = sqrt(10)*randn(1,length(n)); y =
```

```
cumsum(x); ny = n - 5;
[y1,ny1] = sigshift(y,ny,-1); [xs,nxs] = sigshift(x,n,-
1); y2 = cumsum(xs);
ny2 = nxs - 5; [diff,ndiff] = sigadd(y1,ny1,-y2,ny2);
diff = sum(abs(diff));
if (diff < 1e-5)
disp(' *** System-4 is Time-Invariant *** ');
else
disp(' *** System-4 is Time-Varying *** ');
end
```

Matlab verification:

>> *** System-4 is Time-Invariant ***

System-5: $T_5[x(n)] \triangleq y(n) = x(2n)$

1. Analytic determination of time-invariance:

$\mathbf{T_5}[x(n - k)] = x(2n - k) \neq x[2(n - k)] = y(n - k)$

Hence the system $T_5[x(n)]$ is **time-varying**.

2. Matlab script:

```
%% P0212e: To determine whether the system T5[x(n)] =
x(2n) is time-invariant
clear; clc; close all;
n = 0:100; x = sqrt(10)*randn(1,length(n)); [y,ny] =
dnsample(x,n,2);
[y1,ny1] = sigshift(y,ny,1); [xs,nxs] = sigshift(x,n,1);
[y2,ny2] = dnsample(xs,nxs,2); [diff,ndiff] =
sigadd(y1,ny1,-y2,ny2);
diff = sum(abs(diff));
if (diff < 1e-5)
disp(' *** System-5 is Time-Invariant *** ');
else
disp(' *** System-5 is Time-Varying *** ');
end
```

Matlab verification:

>> *** System-5 is Time-Varying ***

System-6: $T_6[x(n)] \triangleq y(n) = \mathrm{round}[x(n)]$

1. Analytic determination of time-invariance:

$\mathbf{T_6}[x(n - k)] = \mathrm{round}[x(n - k)] = y(n - k)$

Hence the system $T_6[x(n)]$ is **time-invariant**.

2. Matlab script:

```
%% P0212f: To determine if the system
T6[x(n)]=round(x(n)) is time-invariant
clear; clc; close all;
n = 0:100; x = sqrt(10)*randn(1,length(n)); y = round(x);
ny = n;
[y1,ny1] = sigshift(y,ny,1); [xs,nxs] = sigshift(x,n,1);
```

```
y2 = round(xs);
ny2 = nxs; [diff,ndiff] = sigadd(y1,ny1,-y2,ny2); diff =
sum(abs(diff));
if (diff < 1e-5)
disp(' *** System-6 is Time-Invariant *** ');
else
disp(' *** System-6 is Time-Varying *** ');
end
```

Matlab verification:

>> *** System-6 is Time-Invariant ***

P2.13

For the systems given in Problem P2.11, determine analytically their stability and causality.

Solutions

System-1: $T_1[x(n)] \triangleq y(n) = x(n)u(n)$: This system is **stable** since $|y(n)| = |x(n)|$. It is also **causal** since the output depends only on the present value of the input.

System-2: $T_2[x(n)]$, $y(n) = x(n) + n\,x(n+1)$: This system is **unstable** since

$|y(n)| \le |x(n)| + |n||x(n+1)| \nearrow \infty$ as $n \nearrow \infty$ for $|x(n)| < \infty$

It is also **noncausal** since the output $y(n)$ depends on the future input value $x(n+1)$ for $n > 0$.

System-3: $T_3[x(n)] \triangleq y(n) = x(n) + \frac{1}{2}x(n-2) - \frac{1}{3}x(n-3)x(2n)$: This system is **stable** since

$|y(n)| \le |x(n)| + \frac{1}{2}|x(n-2)| + \frac{1}{3}|x(n-3)||x(2n)| < \infty$ for $|x(n)| < \infty$

It is however is **noncausal** since $y(1)$ needs $x(2)$ which is a future input value.

System-4: $T_4[x(n)] \triangleq y(n) = \sum_{k=-\infty}^{n+5} 2x(k)$: This system is **unstable** since

$|y(n)| \le 2\sum_{k=-\infty}^{n+5} |x(k)| \nearrow \infty$ as $n \nearrow \infty$ for $|x(n)| < \infty$

It is also **noncausal** since the output $y(n)$ depends on the future input value $x(n+5)$ for $n > 0$.

System-5: $T_5[x(n)] \triangleq y(n) = x(2n)$: This system is **stable** since $|y(n)| = |x(2n)| < \infty$ for $|x(n)| < \infty$. It is however **noncausal** since $y(1)$ needs $x(2)$ which is a future input value.

System-6: $T_6[x(n)] \triangleq y(n) = \mathrm{round}[x(n)]$: This system is **stable** and **causal**.

P2.14

The linear convolution defined in (2.14) has several properties:

$$
\begin{aligned}
&x_1(n) * x_2(n) = x_1(n) * x_2(n) && : \text{Commutation} \\
&[x_1(n) * x_2(n)] * x_3(n) = x_1(n) * [x_2(n) * x_3(n)] && : \text{Association} \\
&x_1(n) * [x_2(n) + x_3(n)] = x_1(n) * x_2(n) + x_1(n) * x_3(n) && : \text{Distribution} \\
&x(n) * \delta(n - n0) = x(n - n0) && : \text{Identity}
\end{aligned}
\tag{2.28}
$$

1. Analytically prove these properties.
2. Using the following three sequences, verify the above properties.

$x_1(n) = \cos(\pi n/4)[u(n + 5) - u(n - 25)]$

$x_2(n) = (0.9)\text{-}n[u(n) - u(n - 20)]$

$x_3(n) = \text{round}[5w(n)]$, $-10 \leq n \leq 10$; where $w(n)$ is uniform over $[-1, 1]$

Use the **conv_m** function.

Solutions

Verification using Matlab:

Commutation Matlab script:

```
% P2.14
%% P0214a: To prove the Commutation property of
convolution
% i.e. conv(x1(n),x2(n)) = conv(x2(n), x1(n))
clear; clc; close all;
n1 = -10:30; n2 = 0:25; n3 = -10:10; x11 = cos(pi*n1/4);
n11 = n1; [x12,n12] = stepseq(-5,-10,30); [x13,n13] =
stepseq(25,-10,30);
[x14,n14] = sigadd(x12,n12,-x13,n13); x1 = x11.*x14;
x21 = 0.9 .^ -n2; [x22,n22] = stepseq(0,0,25); [x23,n23]
= stepseq(20,0,25);
x24 = x22 - x23; x2 = x21.*x24;
x3 = round((rand(1,21)*2 - 1)*5);
% Commutative property
[y1,ny1] = conv_m(x1,n1,x2,n2); [y2,ny2] =
conv_m(x2,n2,x1,n1);
ydiff = max(abs(y1 - y2)), ndiff = max(abs(ny1 - ny2)),
```

Matlab verification:

```
ydiff =
     0
ndiff =
     0
```

Association Matlab script:

```
%% P0214b: To prove the Association property of
convolution
% i.e. conv(conv(x1(n),x2(n)),x3(n)) =
conv(x1(n),conv(x2(n),x3(n)))
clear; clc; close all;
n1 = -10:30; n2 = 0:25; n3 = -10:10; x11 = cos(pi*n1 /
4); n11 = n1;
[x12,n12] = stepseq(-5,-10,30); [x13,n13] = stepseq(25,-
10,30);
```

```
[x14,n14] = sigadd(x12,n12,-x13,n13); x1 = x11.*x14;
x21 = 0.9 .^ -n2; [x22,n22] = stepseq(0,0,25); [x23,n23]
= stepseq(20,0,25);
x24 = x22 - x23; x2 = x21.*x24; x3 = round((rand(1,21)*2
- 1)*5);
% Association property
[y1,ny1] = conv_m(x1,n1,x2,n2); [y1,ny1] =
conv_m(y1,ny1,x3,n3);
[y2,ny2] = conv_m(x2,n2,x3,n3); [y2,ny2] =
conv_m(x1,n1,y2,ny2);
ydiff = max(abs(y1 - y2)), ndiff = max(abs(ny1 - ny2)),
```

Matlab verification:

```
Ydiff  =
      0
ndiff  =
      0
```

Distribution Matlab script:

```
%% P0214c: To prove the Distribution property of
convolution
% i.e.
conv(x1(n),(x2(n)+x3(n)))=conv(x1(n),x2(n))+conv(x1(n),x3
(n))
clear; clc; close all;
n1 = -10:30; n2 = 0:25; n3 = -10:10; x11 = cos(pi*n1 /
4); n11 = n1;
[x12,n12] = stepseq(-5,-10,30); [x13,n13] = stepseq(25,-
10,30);
[x14,n14] = sigadd(x12,n12,-x13,n13); x1 = x11.*x14; x21
= 0.9 .^ -n2;
[x22,n22] = stepseq(0,0,25); [x23,n23] =
stepseq(20,0,25); x24 = x22 - x23;
x2 = x21.*x24; x3 = round((rand(1,21)*2 - 1)*5);
% Distributive property
[y1,ny1] = sigadd(x2,n2,x3,n3); [y1,ny1] =
conv_m(x1,n1,y1,ny1);
[y2,ny2] = conv_m(x1,n1,x2,n2); [y3,ny3] =
conv_m(x1,n1,x3,n3);
[y4,ny4] = sigadd(y2,ny2,y3,ny3); ydiff = max(abs(y1 -
y4)),
ndiff = max(abs(ny1 - ny4)),
```

Matlab verification:

```
ydiff  =
      0
ndiff  =
```

0

Identity Matlab script:

```
%% P0214d: To prove the Identity property of convolution
% i.e. conv(x(n),delta(n - n0)) = x(n - n0)
clear; clc; close all;
n1 = -10:30; n2 = 0:25; n3 = -10:10; x11 = cos(pi*n1 /
4); n11 = n1;
[x12,n12] = stepseq(-5,-10,30); [x13,n13] = stepseq(25,-
10,30);
[x14,n14] = sigadd(x12,n12,-x13,n13); x1 = x11.*x14; x21
= 0.9 .^ -n2;
[x22,n22] = stepseq(0,0,25); [x23,n23] =
stepseq(20,0,25); x24 = x22 - x23;
x2 = x21.*x24; x3 = round((rand(1,21)*2 - 1)*5);
% Identity property
n0 = fix(100*rand(1,1)-0.5); [dl,ndl] = impseq(n0,n0,n0);
dl = double(dl);            % transfer logical format to
double format, or function conv2 will go wrong
[y11,ny11] = conv_m(x1,n1,dl,ndl); [y12,ny12] =
sigshift(x1,n1,n0);
y1diff = max(abs(y11 - y12)), ny1diff = max(abs(ny11 -
ny12)),
[y21,ny21] = conv_m(x2,n2,dl,ndl); [y22,ny22] =
sigshift(x2,n2,n0);
y2diff = max(abs(y21 - y22)), ny2diff = max(abs(ny21 -
ny22)),
[y31,ny31] = conv_m(x3,n3,dl,ndl); [y32,ny32] =
sigshift(x3,n3,n0);
y3diff = max(abs(y31 - y32)), ny3diff = max(abs(ny31 -
ny32)),
```

Matlab verification:

ydiff =

0

ndiff =

0

P2.15

Determine analytically the convolution $y(n) = x(n) * h(n)$ of the following sequences, and verify your answers using the **conv_m** function.

1. $x(n) = \{2, -4, 5, 3, -1, -2, 6\}$, $h(n) = \{1, -1, 1, -1, 1\}$

↑ ↑

2. $x(n) = \{1, 1, 0, 1, 1\}$, $h(n) = \{1, -2, -3, 4\}$

↑ ↑

3. $x(n) = (1/4)\text{-}n[u(n + 1) - u(n - 4)]$, $h(n) = u(n) - u(n - 5)$
4. $x(n) = n/4[u(n) - u(n - 6)]$, $h(n) = 2[u(n + 2) - u(n - 3)]$

Solutions

1. $x(n) = \{2, -4, 5, 3, -1, -2, 6\}$, $h(n) = \{1, -1, 1, -1, 1\}$
↑ ↑

Matlab script:

```
%% P0215a
n1 = -3:3; x = [2 -4 5 3 -1 -2 6]; n2 = -1:3; h = [1 -1 1
-1 1];
[y,n] = conv_m(x,n1,h,n2)
y =
     2    -6    11    -8     7    -7     9    -4     7    -8     6
n =
    -4    -3    -2    -1     0     1     2     3     4     5     6
```

2. $x(n) = \{1, 1, 0, 1, 1\}$, $h(n) = \{1, -2, -3, 4\}$
↑ ↑

Matlab script:

```
%% P0215b
n1 = -3:3; x = [1 1 0 1 1]; n2 = -3:0; h = [1 -2 -3 4];
[y,n] = conv_m(x,n1,h,n2)
y =
     1    -1    -5     2     3    -5     1     4
n =
    -6    -5    -4    -3    -2    -1     0     1
```

3. $x(n) = (1/4)\text{-}n[u(n + 1) - u(n - 4)]$, $h(n) = u(n) - u(n - 5)$

Matlab script:

```
%% P0215c
n1 = -2:5; [x11,nx11] = stepseq(-1,-2,5); [x12,nx12] =
stepseq(4,-2,5);
[x13,n13] = sigadd(x11,nx11,-x12,nx12); x14 = 0.25 .^ -
n1; n14 = n1;
x = x14 .* x13;
n2 = 0:6; [h11,nh11] = stepseq(0,0,6); [h12,nh12] =
stepseq(5,0,6); h=h11-h12;
[y,n] = conv_m(x,n1,h,n2)
y =
    0    0.2500    1.2500    5.2500    21.2500    85.2500    85.0000    84.0000
80.0000    64.0000         0         0         0         0
n =
    -2    -1     0     1     2     3     4     5     6     7     8     9
10    11
```

4. $x(n) = n/4[u(n) - u(n - 6)]$, $h(n) = 2[u(n + 2) - u(n - 3)]$

Matlab script:

```
%% P0215d
n1 = 0:7; [x11,nx11] = stepseq(0,0,7); [x12,nx12] =
stepseq(6,0,7);
[x13,n13] = sigadd(x11,nx11,-x12,nx12); x14 = n1/4; n14 =
n1; x = x14 .* x13;
n2 = -3:4; [h11,nh11] = stepseq(-2,-3,4); [h12,nh12] =
stepseq(3,-3,4);
h = 2 * (h11 - h12); [y,n] = conv_m(x,n1,h,n2)
```

y =

     0   0   0.5000   1.5000   3.0000   5.0000   7.5000
7.0000   6.0000   4.5000   2.5000   0   0   0   0

n =

    -3   -2   -1   0   1   2   3   4   5   6   7   8
9   10   11

P2.16

Let $x(n) = (0.8)^n u(n)$, $h(n) = (-0.9)^n u(n)$, and $y(n) = h(n) * x(n)$. Use 3 columns and 1row of subplots for the following parts.

1. Determine $y(n)$ analytically. Plot first 51 samples of $y(n)$ using the **stem** function.
2. Truncate $x(n)$ and $h(n)$ to 26 samples. Use **conv** function to compute $y(n)$. Plot $y(n)$ using the **stem** function. Compare your results with those of part 1.
3. Using the **filter** function, determine the first 51 samples of $x(n) * h(n)$. Plot $y(n)$ using the **stem** function. Compare your results with those of parts 1 and 2.

Solutions

1. Convolution $y(n) = h(n) * x(n)$:

$y(n) = \sum_{k=-\infty}^{\infty} \mathrm{h(k)x(n-k)} = \sum_{k=0}^{\infty} (-0.9)^{\mathbf{k}} (0.8)^{\mathbf{n-k}} \mathrm{u(n-k)} =$

$\left[\sum_{k=0}^{n} (-0.9)^{k} (0.8)^{n} (0.8)^{-k}\right] u(n) = (0.8)^{\boldsymbol{n}} \left[\sum_{k=0}^{n} \left(-\frac{9}{8}\right)^{k}\right] u(n) = \frac{0.8^{n+1} - (-0.9)^{n+1}}{1.7}$

Matlab script:

```
% P2.16
clc; close all;
% run defaultsettings;
n = [0:50]; x = 0.8.^n; h = (-0.9).^n;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0216');
% (a) Plot of the analytical convolution
y1 = ((0.8).^(n+1) - (-0.9).^(n+1))/(0.8+0.9);
subplot(1,3,1); Hs1 = stem(n,y1,'filled');
```

```
set(Hs1,'markersize',2);
title('Analytical'); xlabel('n'); ylabel('y(n)');
```

2. Computation using convolution of truncated sequences: Matlab script

```
% (b) Plot using the conv function and truncated
sequences
x2 = x(1:26); h2 = h(1:26); y2 = conv(h2,x2);
subplot(1,3,2); Hs2 = stem(n,y2,'filled');
set(Hs2,'markersize',2);
title('Using conv function'); xlabel('n');
```

3. To use the Matlab's **filter** function we have to represent the $h(n)$ sequence by coefficients an equivalent difference equation. Matlab script:

```
% (c) Plot of the convolution using the filter function
y3 = filter([1],[1,0.9],x);
subplot(1,3,3); Hs3 = stem(n,y3,'filled');
set(Hs3,'markersize',2);
title('Using filter function'); xlabel('n');
```

The plots of this solution are shown in Figure 2.32. The analytical solution to the convolution in 1 is the exact answer. In the **filter** function approach of 2, the infinite-duration sequence $x(n)$ is exactly represented by coefficients of an equivalent filter. Therefore, the filter solution should be exact except that it is evaluated up to the length of the input sequence. The truncated-sequence computation in 3 is correct up to the first 26 samples and then it degrades rapidly.

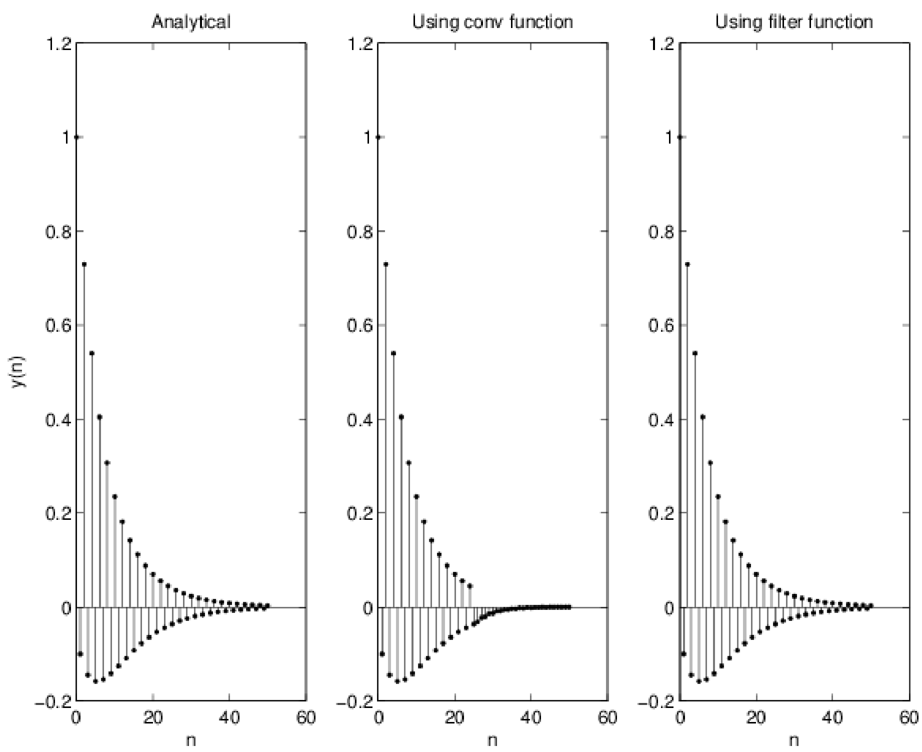


Figure 2.32: Problem P2.16 convolution plots

P2.17

When the sequences $x(n)$ and $h(n)$ are of finite duration $\mathbf{N}_{\boldsymbol{x}}$ and $\mathbf{N}_{\boldsymbol{h}}$, respectively, then their linear convolution (2.13) can also be implemented using *matrix-vector multiplication*. If elements of $y(n)$ and $x(n)$ are arranged in column vectors **x** and **y** respectively, then from (2.13) we obtain

$$\mathbf{y} = \mathbf{Hx}$$

where linear shifts in $h(n - k)$ for $n = 0, \ldots, \mathbf{N}_{\boldsymbol{h}} - 1$ are arranged as rows in the matrix **H**. This matrix has an interesting structure and is called a *Toeplitz* matrix. To investigate this matrix, consider the sequences

$$x(n) = \{\underset{\uparrow}{1}, 2, 3, 4, 5\} \text{ and } h(n) = \{\underset{\uparrow}{6}, 7, 8, 9\}$$

1. Determine the linear convolution $y(n) = h(n) * x(n)$.
2. Express $x(n)$ as a 5×1 column vector **x** and $y(n)$ as a 8×1 column vector **y**. Now determine the 8×5 matrix **H** so that $\mathbf{y} = \mathbf{Hx}$.
3. Characterize the matrix **H**. From this characterization can you give a definition of a Toeplitz matrix? How does this definition compare with that of time invariance?
4. What can you say about the first column and the first row of **H**?

Solutions

1. The linear convolution of the above two sequences is

$$y(n) = \{6, 19, 40, 70, 100, 94, 76, 45\}$$

2. The vector representation of the above operation is:

$$\underbrace{\begin{bmatrix} 6 \\ 19 \\ 40 \\ 70 \\ 100 \\ 94 \\ 76 \\ 45 \end{bmatrix}}_{\mathrm{y}} = \underbrace{\begin{bmatrix} 6 & 0 & 0 & 0 & 0 \\ 7 & 6 & 0 & 0 & 0 \\ 8 & 7 & 6 & 0 & 0 \\ 9 & 8 & 7 & 6 & 0 \\ 0 & 9 & 8 & 7 & 6 \\ 0 & 0 & 9 & 8 & 7 \\ 0 & 0 & 0 & 9 & 8 \\ 0 & 0 & 0 & 0 & 9 \end{bmatrix}}_{\mathrm{H}} \underbrace{\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}}_{x}$$

(a) Note that the matrix **H** has an interesting structure. Each diagonal of **H** contains the same number. Such a matrix is called a Toeplitz matrix. It is characterized by the following property

$$[\mathbf{H}]_{i,j} = [\mathbf{H}]_{i-j}$$

which is similar to the definition of time-invariance.

(b) Note carefully that the first column of **H** contains the impulse response vector $h(n)$ followed by number of zeros equal to the number of $x(n)$ values minus one. The first row contains the first element of $h(n)$ followed by the same number of zeros as in the first column. Using this information and the above property we can generate the whole Topelitz matrix.

P2.18

MATLAB provides a function called **toeplitz** to generate a Toeplitz matrix, given the first row and the first column.

1. Using this function and your answer to Problem P2.17, part 4, develop another MATLAB function to implement linear convolution. The format of the function should be

function [y,H]=conv_tp(h,x)
% Linear Convolution using Toeplitz Matrix
% --
% [y,H] = conv_tp(h,x)
% y = output sequence in column vector form
% H = Toeplitz matrix corresponding to sequence h so that y = Hx
% h = Impulse response sequence in column vector form
% x = input sequence in column vector form

2. Verify your function on the sequences given in Problem P2.17.

Solutions

1. The Matlab function **conv_tp**:

```
function [y,H]=conv_tp(h,x)
% Linear Convolution using Toeplitz Matrix
% ----------------------------------------
% [y,H] = conv_tp(h,x)
% y = output sequence in column vector form
% H = Toeplitz matrix corresponding to sequence h so that
y = Hx
% h = Impulse response sequence in column vector form
% x = input sequence in column vector form
%
Nx = length(x);Nh = length(h);
hc = [h;zeros(Nx-1,1)];
hr = [h(1),zeros(1,Nx-1)];
H = toeplitz(hc,hr); y = H*x;
```

2. Matlab verification

```
% P2.18
x = [1,2,3,4,5]'; h = [6,7,8,9]';
[y,H] = conv_tp(h,x); y = y',H
```

```
y =
     6    19    40    70   100    94    76    45
```

H =

6	0	0	0	0
7	6	0	0	0
8	7	6	0	0
9	8	7	6	0
0	9	8	7	6
0	0	9	8	7
0	0	0	9	8
0	0	0	0	9

P2.19

A linear and time-invariant system is described by the difference equation

$$y(n) - 0.5y(n-1) + 0.25y(n-2) = x(n) + 2x(n-1) + x(n-3)$$

1. Using the **filter** function, compute and plot the impulse response of the system over $0 \le n \le 100$.
2. Determine the stability of the system from this impulse response.
3. If the input to this system is $x(n) = [5 + 3\cos(0.2\pi n) + 4\sin(0.6\pi n)]\, u(n)$, determine the response $y(n)$ over $0 \le n \le 200$ using the **filter** function.

Solutions

(a) Impulse response using the Using the **filter** function

```
% P2.19
%% P0219a: System response using the filter function
clc; close all;
a = [1,-0.5,0.25];b = [1,2,0,1];[delta,n] =
impseq(0,0,100);
h = filter(b,a,delta);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0219a');
Hs = stem(n,h,'filled'); set(Hs,'markersize',2);
axis([min(n)-5,max(n)+5,min(h)-0.5,max(h)+0.5]);
xlabel('n','FontSize',12); ylabel('h(n)','FontSize',12);
title('Impulse response','FontSize',12);
print -deps2 ../EPSFILES/P0219a;
```

The plots of the impulse response $h(n)$ is shown in Figure 2.33

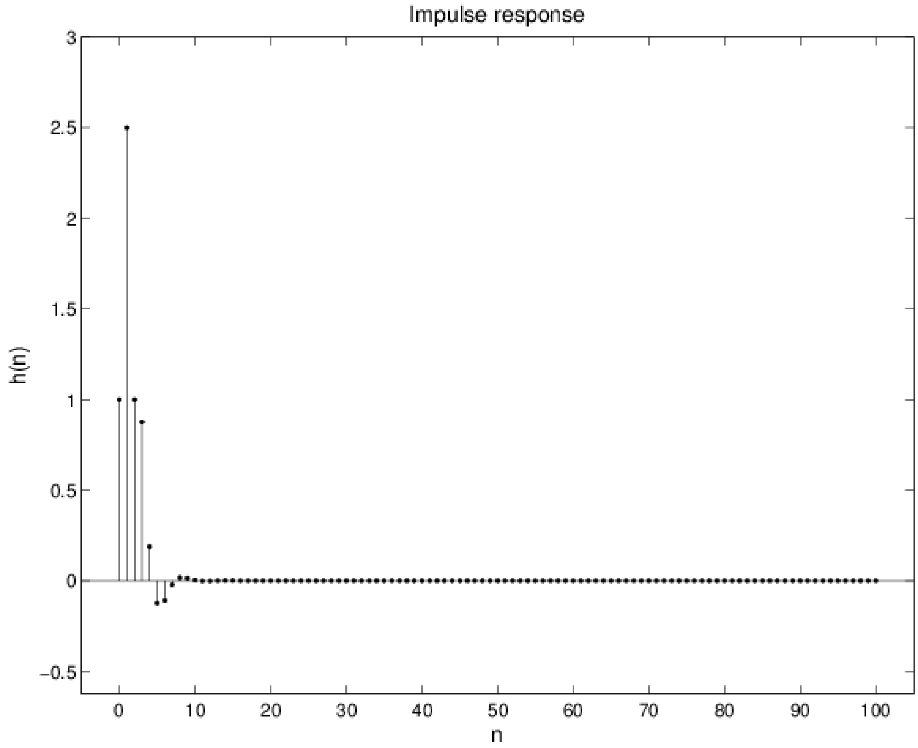


Figure 2.33: Problem P2.19.1 impulse response plot

(b) Clearly from Figure 2.33 the system is stable.

(c) Response $y(n)$ when the input is $x(n) = [5 + 3\cos(0.2\pi n) + 4\sin(0.6\pi n)]\,u(n)$:

```
%% P0219c: Output response of a system using the filter
function.
clc; close all;
b = [1 2 0 1]; a = [1 -0.5 0.25]; n = 0:200;
x = 5*ones(size(n))+3*cos(0.2*pi*n)+4*sin(0.6*pi*n); y =
filter(b,a,x);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0219c');
Hs = stem(n,y,'filled'); set(Hs,'markersize',2); axis([-
10,210,0,50]);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
title('Output response','FontSize',12);
print -deps2 ../EPSFILES/P0219c;
```

The plots of the response $y(n)$ is shown in Figure 2.34.

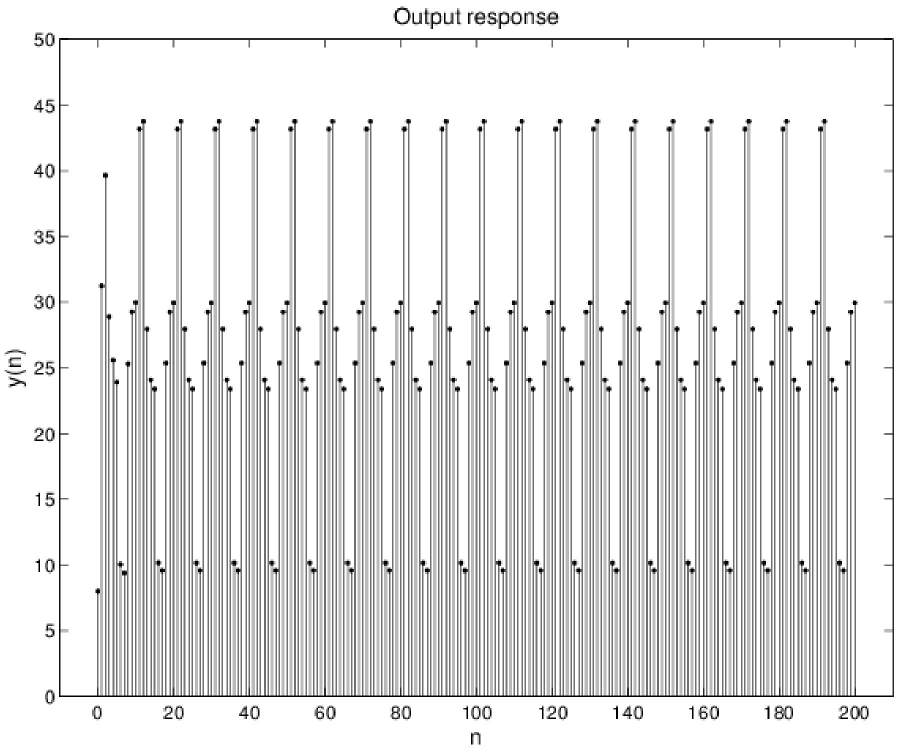


Figure 2.34: Problem P2.19.3 response plot

P2.20

A "simple" *digital differentiator* is given by

$$y(n) = x(n) - x(n - 1)$$

which computes a backward first-order difference of the input sequence. Implement this differentiator on the following sequences, and plot the results. Comment on the appropriateness of this simple differentiator.

1. $x(n) = 5\,[u(n) - u(n - 20)]$: a rectangular pulse
2. $x(n) = n[u(n) - u(n - 10)] + (20 - n)\,[u(n - 10) - u(n - 20)]$: a triangular pulse
3. $x(n) = \sin\left(\frac{\pi n}{25}\right)\,[u(n) - u(n - 100)]$: a sinusoidal pulse

Solutions

(a) Response to a rectangular pulse $x(n) = 5\,[u(n) - u(n - 20)]$:

```
% P2.20
%% P0220a: Simple Differentiator response to a
rectangular pulse
clc; close all;
a = 1; b = [1 -1]; n1 = 0:22;
[x11,nx11] = stepseq(0,0,22); [x12,nx12] =
```

```
stepseq(20,0,22);
x1 = 5*(x11 - x12); y1 = filter(b,a,x1);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0220a');
Hs = stem(n1,y1,'filled'); set(Hs,'markersize',2);
axis([-1,23,-6,6]);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
ytick = [-6:6];
title('Output response for rectangular pulse
','FontSize',12);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0220a;
```

The plots of the response $y(n)$ is shown in Figure 2.35.

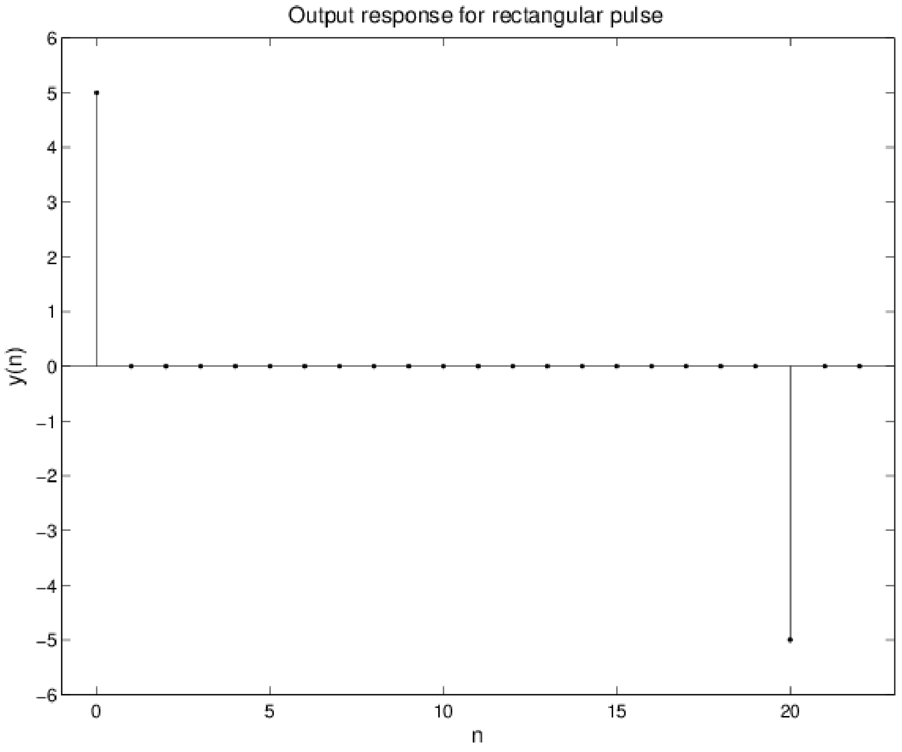


Figure 2.35: Problem P2.20.1 response plot

(b) Response to a triangular pulse $x(n) = n\,[u(n) - u(n - 10)] + (20 - n)\,[u(n - 10) - u(n - 20)]$:

```
%% P0220b: Simple Differentiator response to a triangular
pulse
clc; close all;
a = 1; b = [1 -1]; n2 = 0:21; [x11,nx11] =
stepseq(0,0,21);
[x12,nx12] = stepseq(10,0,21); [x13,nx13] =
stepseq(20,0,21);
x2 = n2.*(x11 - x12) + (20 - n2).*(x12 - x13); y2 =
```

```
filter(b,a,x2);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0220b');
Hs = stem(n2,y2,'filled'); set(Hs,'markersize',2);
axis([min(n2)-1,max(n2)+1,min(y2)-0.5,max(y2) + 0.5]);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
title('Output response for triangular
pulse','FontSize',12);
print -deps2 ../EPSFILES/P0220b;
```

The plots of the response $y(n)$ is shown in Figure 2.36

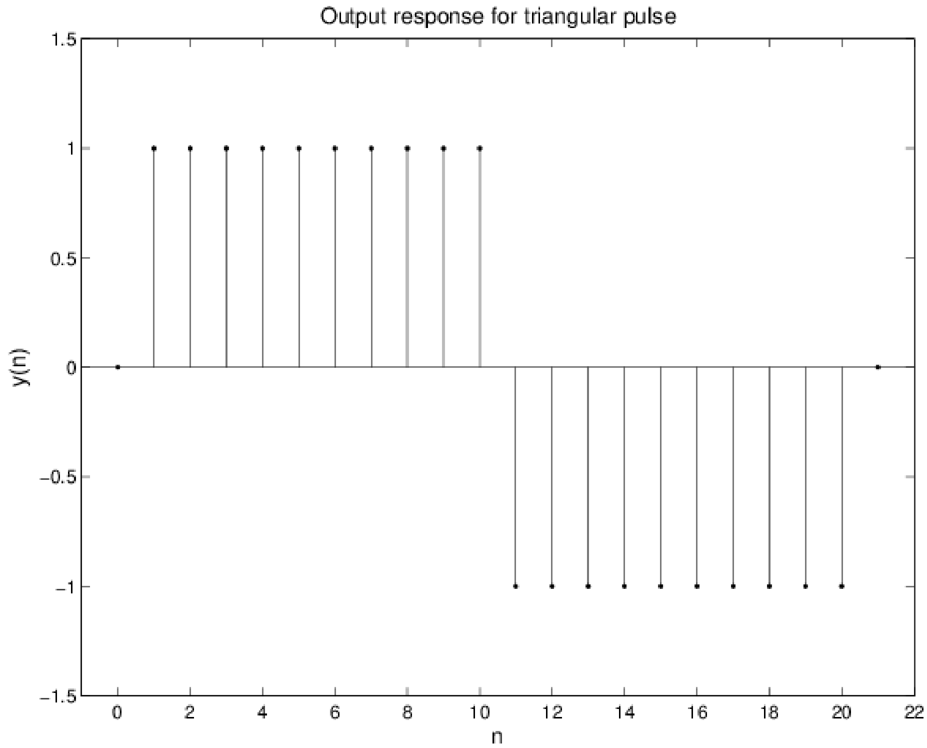


Figure 2.36: Problem P2.20.2 response plot

(c) Response to a sinusoidal pulse $x(n) = \sin\left(\frac{\pi n}{25}\right) [u(n) - u(n - 100)]$:

```
%% P0220cSimple Differentiator response to a sinusoidal
pulse
clc; close all;
a = 1; b = [1 -1]; n3 = 0:101; [x11,nx11] =
stepseq(0,0,101);
[x12,nx12] = stepseq(100,0,101); x13 = x11-x12; x3 =
sin(pi*n3/25).*x13;
y3 = filter(b,a,x3);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0220c');
```

```
Hs = stem(n3,y3,'filled'); set(Hs,'markersize',2);
axis([-5,105,-0.15,0.15]); ytick = [-0.15:0.05:0.15];
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
title('Output response for sinusoidal
pulse','FontSize',12);
set(gca,'YTickMode','manual','YTick',ytick);
print -deps2 ../EPSFILES/P0220c;
```

The plots of the response $y(n)$ is shown in Figure 2.37.

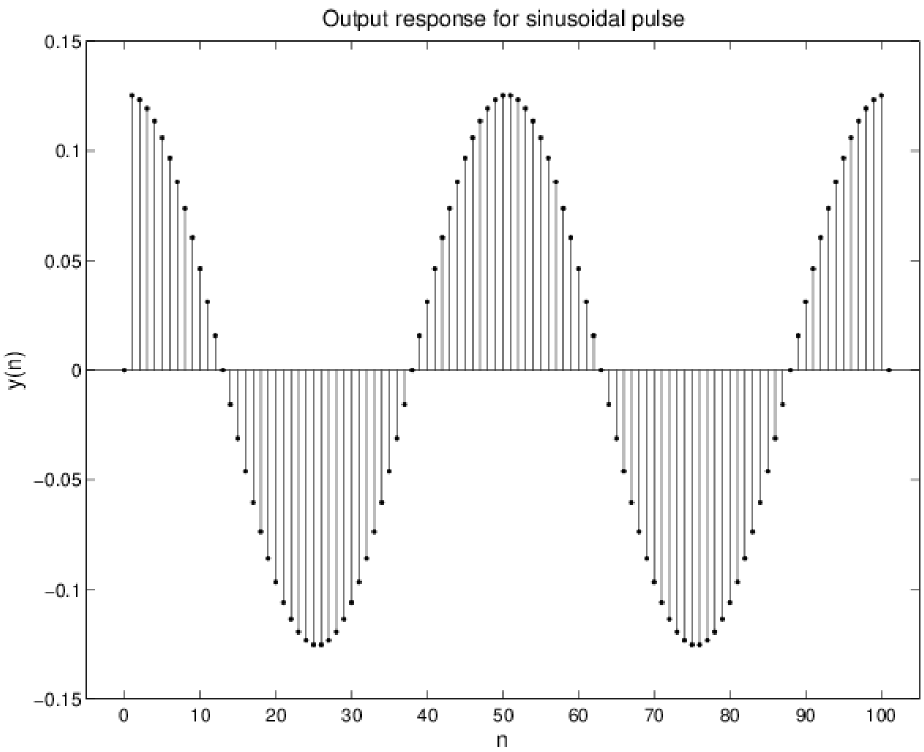


Figure 2.37: Problem P2.20.3 response plot

Chapter 3

P3.1

Using the matrix-vector multiplication approach discussed in this chapter, write a MATLAB function to compute the DTFT of a finite-duration sequence. The format of the function should be

function [X] = dtft(x,n,w)
% Computes Discrete-time Fourier Transform
% [X] = dtft(x,n,w)
% X = DTFT values computed at w frequencies
% x = finite duration sequence over n
% n = sample position vector
% w = frequency location vector

Use this function to compute the DTFT $X(e^{jw})$ of the following finite-duration sequences over $-\pi \leq \omega \leq \pi$. Plot DTFT magnitude and angle graphs in one figure window.

1. $x(n) = (0.6)^{|n|}\ [u(n + 10) - u(n - 11)]$. Comment on the angle plot.
2. $x(n) = n(0.9)^{n}\ [u(n) - u(n - 21)]$.
3. $x(n) = [\cos(0.5\pi n) + j \sin(0.5\pi n)][u(n) - u(n - 51)]$. Comment on the magnitude plot.
4. $x(n) = \{4, \underset{\uparrow}{3}, 2, 1, 1, 2, 3, 4\}$. Comment on the angle plot.
5. $x(n) = \{4, \underset{\uparrow}{3}, 2, 1, -1, -2, -3, -4\}$. Comment on the angle plot.

Solutions

Matlab Function [X] = dtft(x,n,w)

```
function [X] = dtft(x,n,w)
% Computes Discrete-time Fourier Transform
% [X] = dtft(x,n,w)
%
% X = DTFT values computed at w frequencies
% x = finite duration sequence over n (row vector)
% n = sample position row vector
% w = frequency row vector, w = k*pi/M, k = 0:M
X = x*exp(-1i*n'*w);
```

1. $x(n) = (0.6)^{|n|}\ [u(n + 10) - u(n - 11)]$.

```
% P3.1
%% P0301a: DTFT of x1(n) = 0.6 ^ |n|*(u(n+10)-u(n-11))
```

```
clc; close all;
%
[x11,n11] = stepseq(-10,-11,11); [x12,n12] = stepseq(11,-
11,11);
[x13,n13] = sigadd(x11,n11,-x12,n12); n1 = n13; x1 =
(0.6 .^ abs(n1)).*x13;
w1 = linspace(-pi,pi,201); X1 = dtft(x1,n1,w1);
magX1 = abs(X1); phaX1 = angle(X1);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0301a');
subplot(2,1,1); plot(w1/pi,magX1,'LineWidth',1.5);
axis([-1 1 0 4.5]); wtick = [-1:0.2:1]; magtick =
[0:0.5:4.5];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',magtick);
subplot(2,1,2); plot(w1/pi,phaX1*180/pi,'LineWidth',1.5);
axis([-1,1,-180,180]); phatick = [-180 0 180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0301a;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.1.

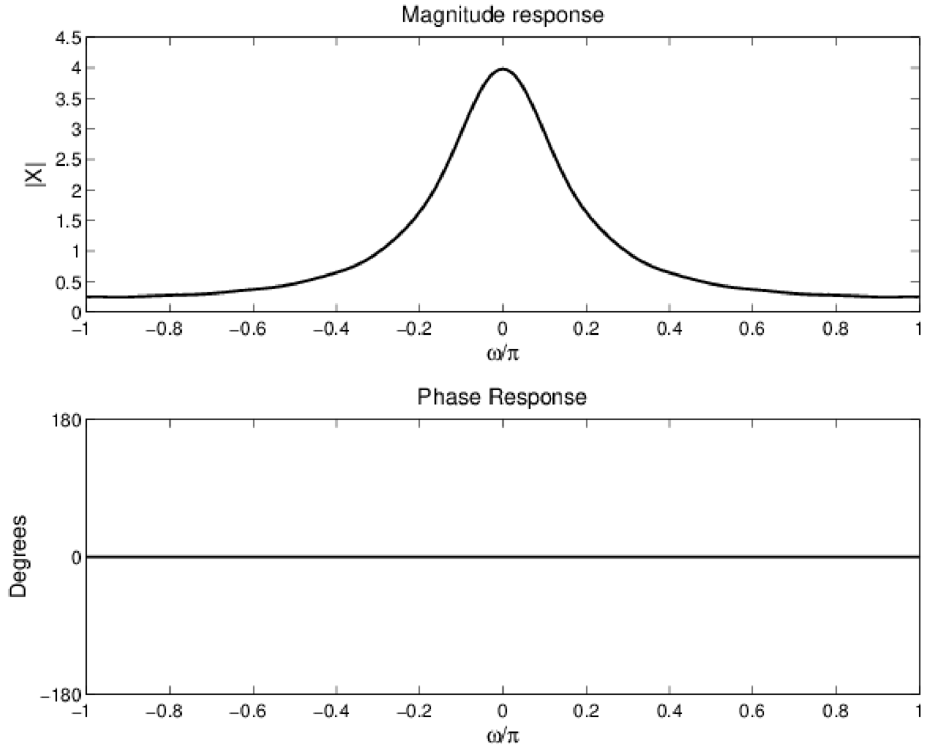


Figure 3.1: Problem P3.1.1 DTFT plots

2. $x(n) = n(0.9)^n \; [u(n) - u(n - 21)]$.

```
%% P0301b: % DTFT of x2(n) = n.*(0.9 ^ n) .*(u(n)-u(n-
21))
clc; close all;
%
[x21,n21] = stepseq(0,0,22); [x22,n22] =
stepseq(21,0,22);
[x23,n23] = sigadd(x21,n21,-x22,n22); n2 = n23; x2 =
n2.*(0.9 .^ n2).*x23;
w2 = linspace(-pi,pi,201); X2 = dtft(x2,n2,w2);
magX2 = abs(X2); phaX2 = angle(X2);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0301b');
subplot(2,1,1); plot(w2/pi,magX2,'LineWidth',1.5);
wtick = [-1:0.2:1]; magtick = [0:10:60];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',magtick);
subplot(2,1,2); plot(w2/pi,phaX2*180/pi,'LineWidth',1.5);
axis([-1,1,-200,200]); phatick = [-180:60:180];
```

```
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0301b;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.2.

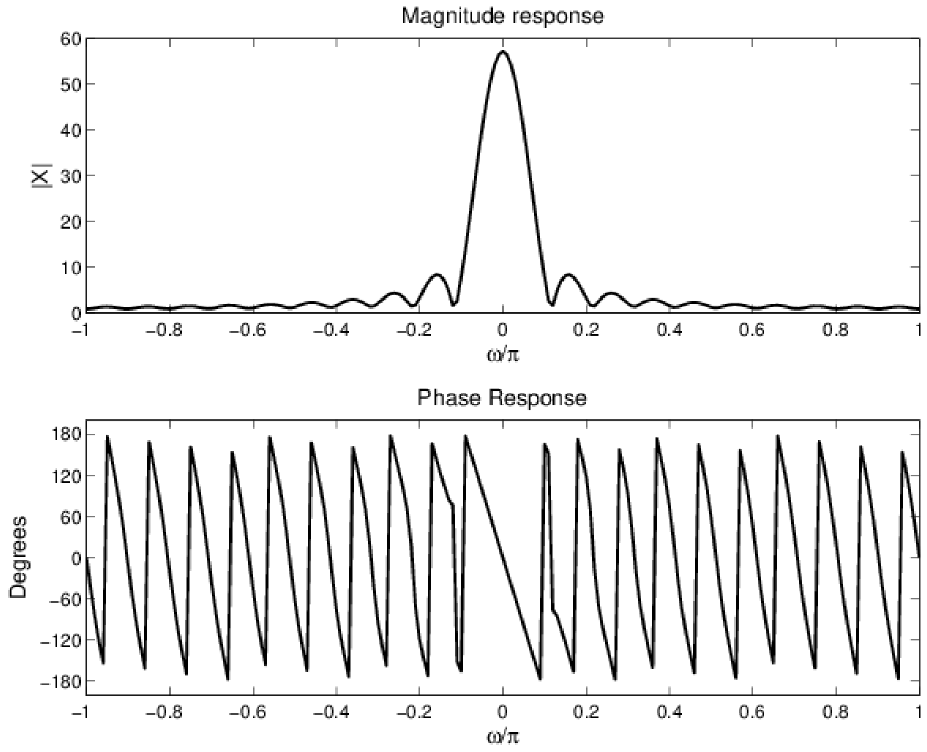


Figure 3.2: Problem P3.1.2 DTFT plots

3. $x(n) = [\cos(0.5\pi n) + j\sin(0.5\pi n)]\,[u(n) - u(n - 51)]$

```
%% P0301c: % DTFT of x3(n) =
(cos(0.5*pi*n)+j*sin(0.5*pi*n)).*(u(n)-u(n-51))
clc; close all;
%
[x31,n31] = stepseq(0,0,52); [x32,n32] =
stepseq(51,0,52);
[x33,n33] = sigadd(x31,n31,-x32,n32); n3 = n33;
x3 = (cos(0.5*pi*n3)+1i*sin(0.5*pi*n3)).*x33;
w3 = linspace(-pi,pi,201); X3 = dtft(x3,n3,w3);
magX3 = abs(X3); phaX3 = angle(X3);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0301c');
subplot(2,1,1); plot(w3/pi,magX3,'LineWidth',1.5);
```

```
wtick = [-1:0.2:1]; magtick = [0:10:60];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',magtick);
subplot(2,1,2,'LineWidth',1.5); plot(w3/pi,phaX3*180/pi);
axis([-1,1,-200,200]); phatick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0301c;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.3.

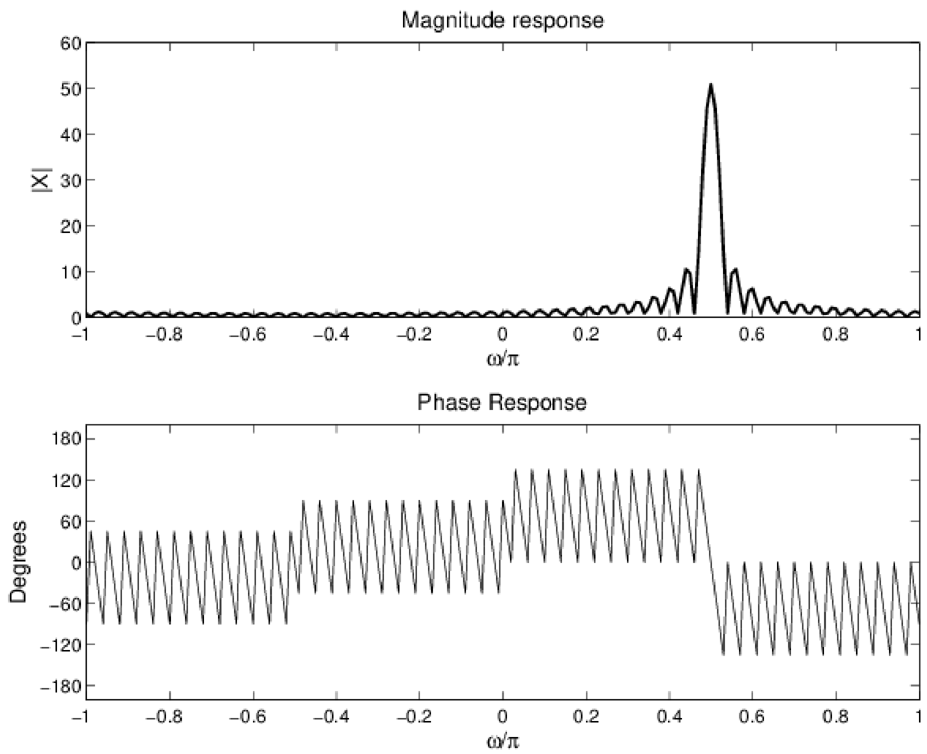


Figure 3.3: Problem P3.1.3 DTFT plots

4. $x(n) = \{4, 3, 2, 1, 1, 2, 3, 4\}$.
$\uparrow$

```
%% P0301d: % DTFT of x4(n) = [4 3 2 1 1 2 3 4] ; n = 0:7;
clc; close all;
%
x4 = [4 3 2 1 1 2 3 4]; n4 = [0:7];
w4 = linspace(-pi,pi,201); X4 = dtft(x4,n4,w4);
```

```
magX4 = abs(X4); phaX4 = angle(X4);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0301d');
subplot(2,1,1); plot(w4/pi,magX4,'LineWidth',1.5);
axis([-1,1,0,25]); wtick = [-1:0.2:1]; magtick =
[0:5:25];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',magtick);
subplot(2,1,2); plot(w4/pi,phaX4*180/pi,'LineWidth',1.5);
axis([-1,1,-200,200]); phatick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0301d;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.4.

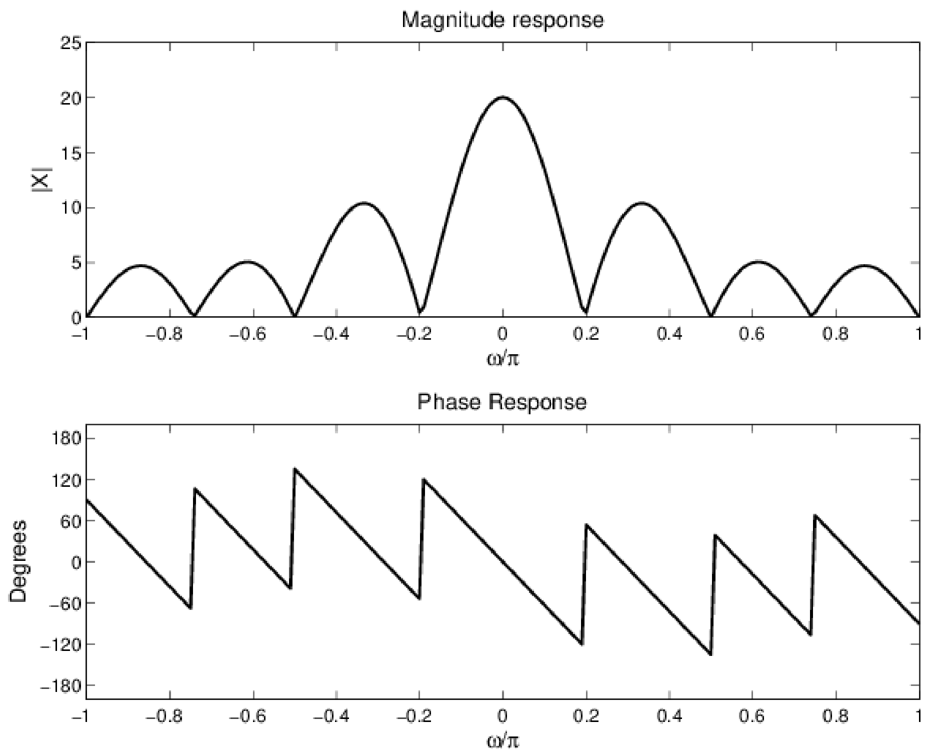


Figure 3.4: Problem P3.1.4 DTFT plots

5. $x(n) = \{4, 3, 2, 1, -1, -2, -3, -4\}$. Comment on the angle plot.

↑
```
%% P0301e: % DTFT of x5(n) = [4 3 2 1 -1 -2 -3 -4] ; n = 0:7;
clc; close all;
%
x5 = [4 3 2 1 -1 -2 -3 -4]; n5 = [0:7];
w5 = linspace(-pi,pi,201); X5 = dtft(x5,n5,w5);
magX5 = abs(X5); phaX5 = angle(X5);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0301e');
subplot(2,1,1); plot(w5/pi,magX5,'LineWidth',1.5);
wtick = [-1:0.2:1]; magtick = [0:5:20]; axis([-1 1 0 20]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',magtick);
subplot(2,1,2); plot(w5/pi,phaX5*180/pi,'LineWidth',1.5);
axis([-1,1,-200,200]); phatick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0301e;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.5.

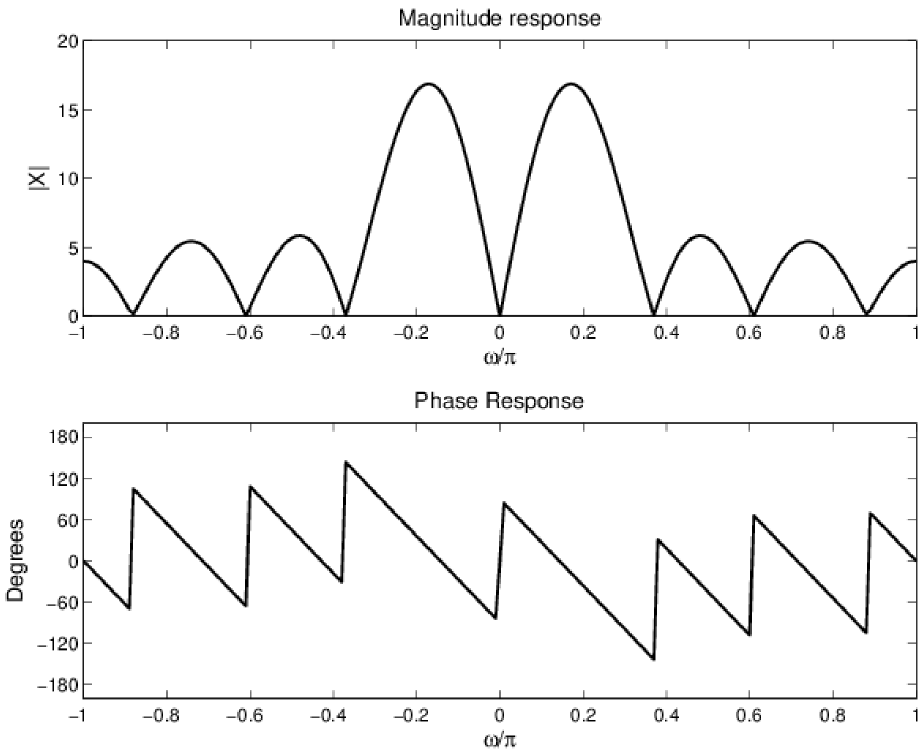


Figure 3.5: Problem P3.1.5 DTFT plots

P3.2

Let $x_1(n) = \{1, 2, 2, 1\}$. A new sequence $x_2(n)$ is formed using
$\uparrow$

$$\mathbf{x_2}(n) = \begin{cases} x_1(n), 0 \le n \le 3; \\ x_1(n-4), 4 \le n \le 7; \\ 0, Otherwise. \end{cases} \qquad \textbf{(3.44)}$$

1. Express $X_2(e^{jw})$ in terms of $X_1(e^{jw})$ without explicitly computing $X_1(e^{jw})$.
2. Verify your result using MATLAB by computing and plotting magnitudes of the respective DTFTs.

Solutions

1. Clearly, $x_2(n) = x_1(n) + x_1(n - 4)$. Hence

$$X_2(e^{jw}) = X_1(e^{jw}) + X_1(e^{jw})e^{-j4w} = 2e^{-j2w}\cos(2\omega)\, X_1(e^{jw})$$

Thus the magnitude $|X_1(e^{jw})|$ is scaled by 2 and changed by $|\cos(2\omega)|$ while the phase of $|X_1(e^{jw})|$ is changed by 2ω.

2. Matlab verification:

```
% P3.2
%% P0302b: x1(n) = [1 2 2 1],n = [0:3];
% x2(n) = x1(n) ,n = [0:3];
```

```
% = x1(n-4) ,n = [4:7];
clc; close all;
n1 = [0:3]; x1 = [1 2 2 1]; n2 = [0:7]; x2 = [x1 x1];
w2 = linspace(-pi,pi,201); X1 = dtft(x1,n1,w2); X2 =
dtft(x2,n2,w2);
magX1 = abs(X1); phaX1 = angle(X1); magX2 = abs(X2);
phaX2 = angle(X2);
wtick = [-1:0.5:1]; phatick = [-180:60 :180];
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0302b');
subplot(2,2,1); plot(w2/pi,magX1,'LineWidth',1.5);
axis([-1 1 0 8]); magtick1 = [0:2:8];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X_1|','FontSize',12);
title(['Magnitude response' char(10) 'signal
x_1'],'FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',magtick1);
subplot(2,2,3); plot(w2/pi,phaX1*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]);
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase response' char(10) 'signal
x_1'],'FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
subplot(2,2,2); plot(w2/pi,magX2,'LineWidth',1.5);
axis([-1 1 0 16]); magtick2 = [0:4:16];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X_2|','FontSize',12);
title(['Magnitude response' char(10) 'signal
x_2'],'FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',magtick2);
subplot(2,2,4); plot(w2/pi,phaX2*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]);
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase response' char(10) 'signal
x_2'],'FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0302b;
```

The magnitude and phase plots of $X_1(e^{jw})$ and $X_2(e^{jw})$ are shown in Figure 3.6 which confirms the observation in part 1. above.

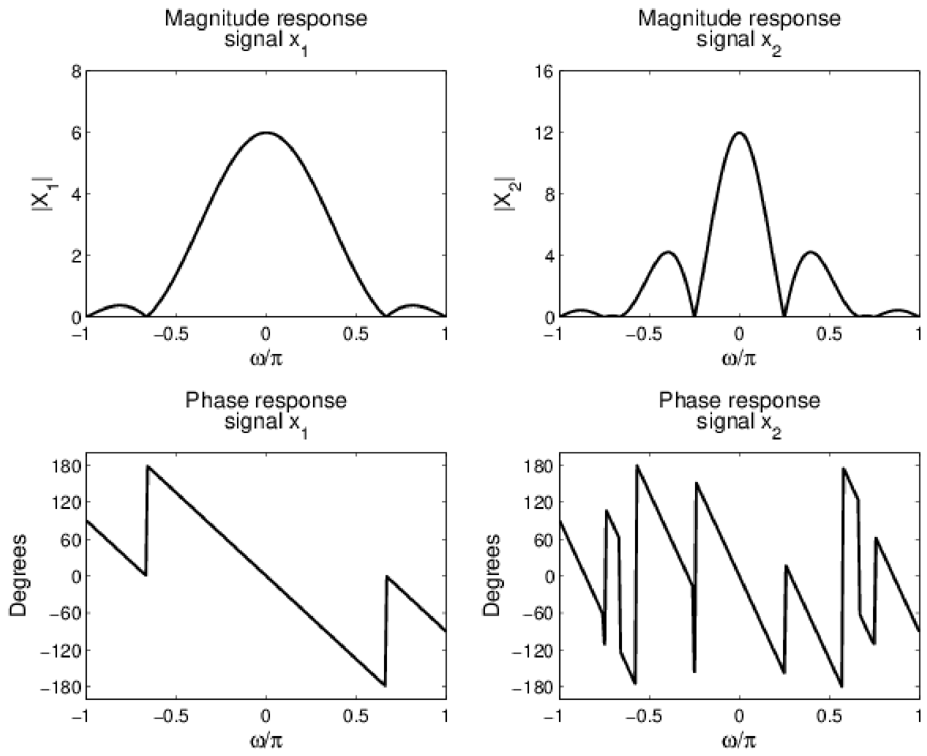


Figure 3.6: Problem P3.2.2 DTFT plots

P3.3

Determine analytically the DTFT of each of the following sequences. Plot the magnitude and angle of $X(e^{jw})$ over $0 \le \omega \le \pi$.

1. $x(n) = 2\ (0.5)^n\ u(n+2)$.
2. $x(n) = \ (0.6)^{|\mathbf{n}|}\ [u(n+10) - u(n-11)]$.
3. $x(n) = n(0.9)^n\ u(n+3)$.
4. $x(n) = (n+3)\ \ (0.8)^{n-1} u(n-2)$.
5. $x(n) = 4\ \ (-0.7)^n\ \cos(0.25\pi n)u(n)$.

Solutions

1. $x(n) = 2(0.5)_n\, u(n+2)$.

$X\left(e^{\,jw}\right) = 2\sum_{-\infty}^{\infty} 0.5^n u(n\ +\ 2)e^{-jnw} = 2\sum_{-2}^{\infty} 0.5^n e^{-jnw} = 2(0.5)^{-2} e^{\,j2w} \sum_0^{\infty} 0.5^n e^{-jnw} =$

$8\ \frac{e^{\,j2w}}{1-0.5e^{-jw}}$

Matlab Verification:

```
% P3.3
%% P0303a: DTFT of x1(n) = 2*((0.5)^n)*u(n+2) =
```

```
% 8*exp(j*2*w)/(1-0.5*exp(-j*w))
clc; close all;
w1 = linspace(0,pi,501); X1 = 8*exp(1i*2*w1)./(1-
0.5*exp(-1i*w1));
magX1 = abs(X1); phaX1 = angle(X1);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0303a');
subplot(2,1,1); plot(w1/pi,magX1,'LineWidth',1.5);
wtick = [0:0.2:1]; magtick = [0:4:20]; axis([0,1,0,20]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w1/pi,phaX1*180/pi,'LineWidth',1.5);
axis([0,1,-200,200]); phatick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0303a
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.7.

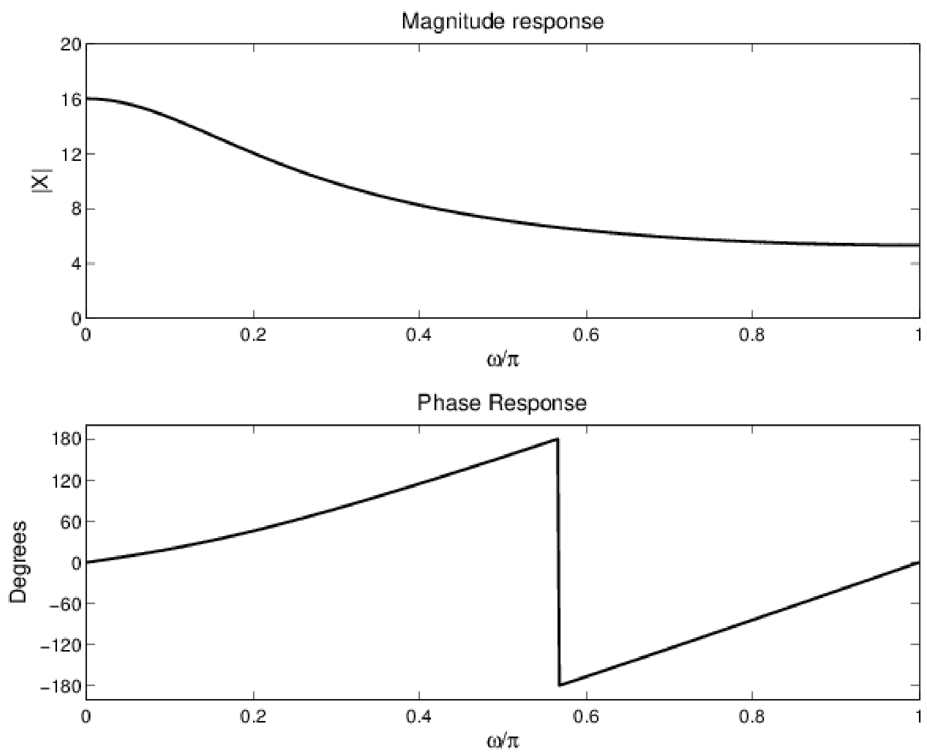


Figure 3.7: Problem P3.3.1 DTFT plots

2. $x(n) = (0.6)^{|n|}\,[u(n + 10) - u(n - 11)]$.

$X(e^{jw})=\sum_{-\infty}^{\infty}(0.6)^{|n|}\,[u(n+10)-u(n-11)]\,e^{-jnw}=\sum_{-10}^{10}0.6^{|n|}e^{-jnw}=$
$\sum_{-10}^{0}0.6^{-n}e^{-jnw}+\sum_{0}^{10}0.6^{n}e^{-jnw}-1=\frac{0.64-2(0.6)^{11}\cos(11\omega)+2(0.6)^{12}\cos(10\omega)}{1.36-1.2\cos(\omega)}$

Matlab Verification:

```
%% P0303b: DTFT of x2(n) = (0.6) ^ |n|*[u(n+10)-u(n-11)]
clc; close all;
w2 = linspace(0,pi,501);
X2 = (0.64-
2*(0.6)^11*cos(11*w2)+2*(0.6)^12*cos(10*w2))./(1.36-
1.2*cos(w2));
magX2 = abs(X2); phaX2 = angle(X2);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0303b');
subplot(2,1,1); plot(w2/pi,magX2,'LineWidth',1.5);
axis([0,1,0,5]); wtick = [0:0.2:1]; magtick = [0:1:5];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w2/pi,phaX2*180/pi,'LineWidth',1.5);
axis([0,1,-200,200]); phatick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0303b;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.8.

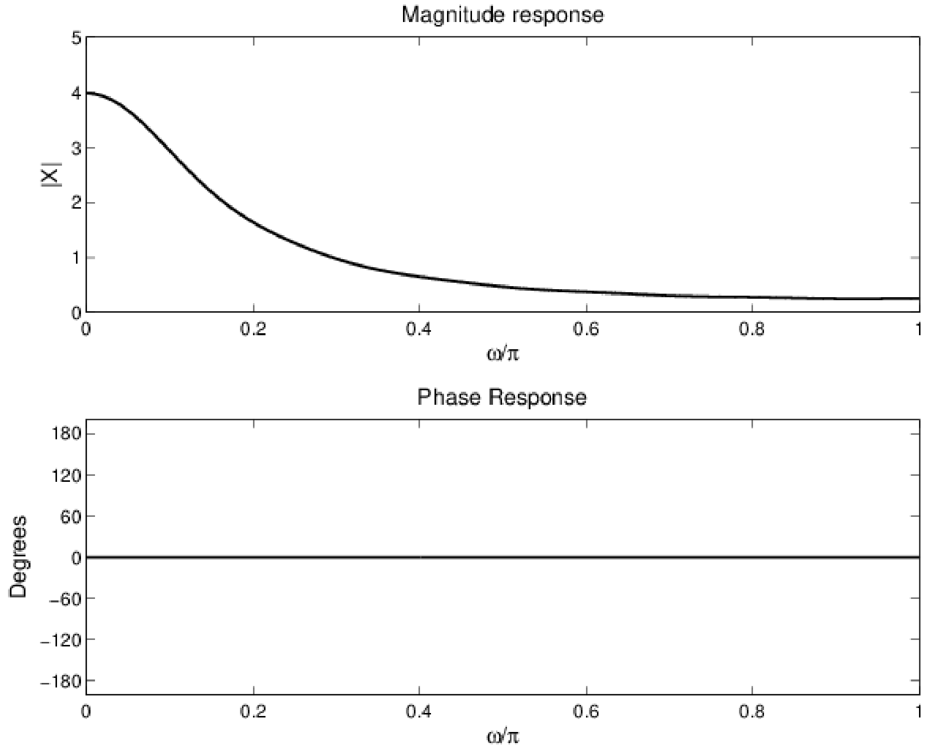


Figure 3.8: Problem P3.3.2 DTFT plots

3. $x(n) = n\,(0.9)^n\,u(n+3)$.

$$X(e^{j\omega}) = \sum_{-\infty}^{\infty} n\,(0.9)^n\,u(n+3)e^{-jn\omega} = \sum_{-3}^{\infty} n\,(0.9)^n\,e^{-jn\omega}$$

$$= -3(0.9)^{-3}e^{j3\omega} - 2(0.9)^{-2}e^{j2\omega} - (0.9)^{-1}e^{j1\omega} + \sum_{0}^{\infty} n\,(0.9)^n\,e^{-jn\omega}$$

$$= -4.1152e^{j3\omega} - 2.4691e^{j2\omega} - 1.1111e^{j\omega} + \frac{0.9e^{-j\omega}}{(1-0.9e^{-j\omega})^2} = \frac{-4.1151e^{j3\omega} + 4.9383e^{j2\omega}}{1 - 1.8e^{-j\omega} + 0.81e^{-j2\omega}}$$

Matlab Verification:

```
%% P0303c: DTFT of x3(n) = n*((0.9) ^ n)*u(n+3);
clc; close all;
w3 = linspace(0,pi,501); X3_num = (-
4.1151*exp(1i*3*w3)+4.9383*exp(1i*2*w3));
X3_den = 1-1.8*exp(-1i*w3)+0.81*exp(-1i*2*w3); X3 =
X3_num./X3_den;
magX3 = abs(X3); phaX3 = angle(X3);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0303c');
subplot(2,1,1); plot(w3/pi,magX3,'LineWidth',1.5);
axis([0,1,0,100]); wtick = [0:0.2:1]; magtick =
[0:20:100];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
```

```
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w3/pi,phaX3*180/pi,'LineWidth',1.5);
axis([0,1,-200,200]); phatick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0303c;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.9.

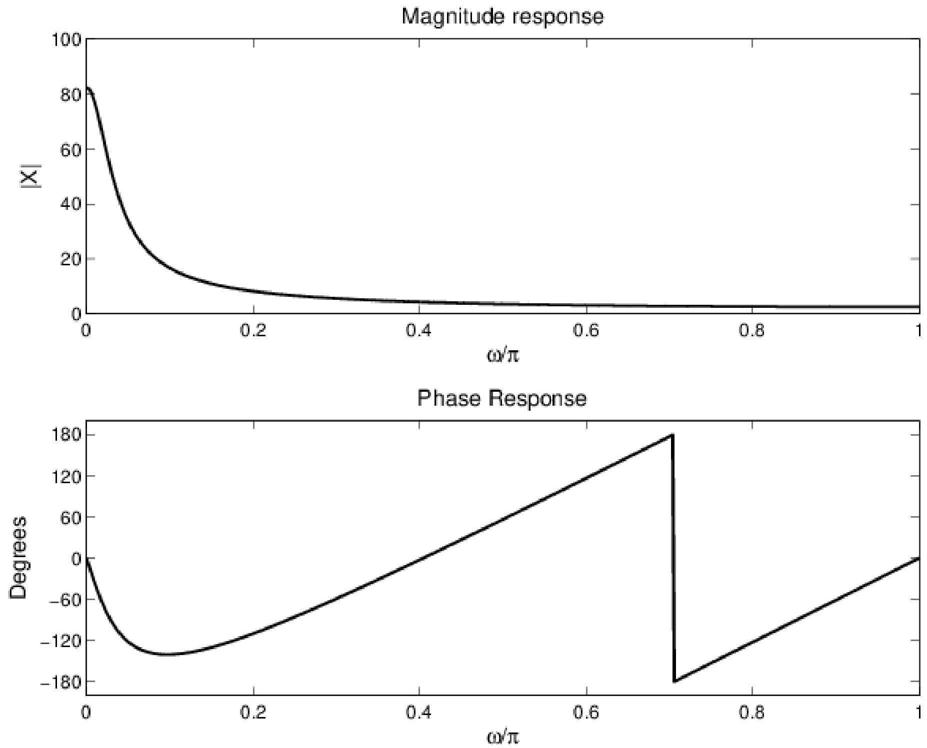


Figure 3.9: Problem P3.3.3 DTFT plots

4. $x(n) = \sum_{-\infty}^{\infty} (n+3)\,(0.8)^{n-1}\,u(n-2)$.

$$X(e^{j\omega}) = \sum_{-\infty}^{\infty} (n+3)\,(0.8)^{n-1}\,u(n-2)e^{-jn\omega} = \sum_{-\infty}^{\infty} (n+5)\,(0.8)^{n+1}\,u(n)e^{-j(n+2)\omega}$$

$$= (0.8)e^{-j2\omega}\sum_{0}^{\infty} n(0.8)^n e^{-jn\omega} + 4e^{-j2\omega}\sum_{0}^{\infty}(0.8)^n e^{-jn\omega}$$

$$= \frac{0.64e^{-j3\omega}}{\left(1-0.8e^{-j\omega}\right)^2} + \frac{4e^{-j2\omega}}{1-0.8e^{-j\omega}} = \frac{4e^{-j2\omega} - 2.56e^{-j3\omega}}{1-1.6e^{-j\omega}+0.64e^{-j2\omega}}$$

Matlab Verification:

```
%% P0303d: DTFT of x4(n) = (n+3)*((0.8) ^ (n-1))*u(n-2);
```

```
clc; close all;
w4 = linspace(0,pi,501); X4_num = 4*exp(-2*1i*w4)-
2.56*exp(-3*1i*w4);
X4_den = 1-1.6*exp(-1*1i*w4)+0.64*exp(-2*1i*w4); X4 =
X4_num./X4_den;
magX4 = abs(X4); phaX4 = angle(X4);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0303d');
subplot(2,1,1); plot(w4/pi,magX4,'LineWidth',1.5);
axis([0 1 0 40]); wtick = [0:0.2:1]; magtick = [0:5:40];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w4/pi,phaX4*180/pi,'LineWidth',1.5);
axis([0,1,-200,200]); phatick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0303d;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.10.

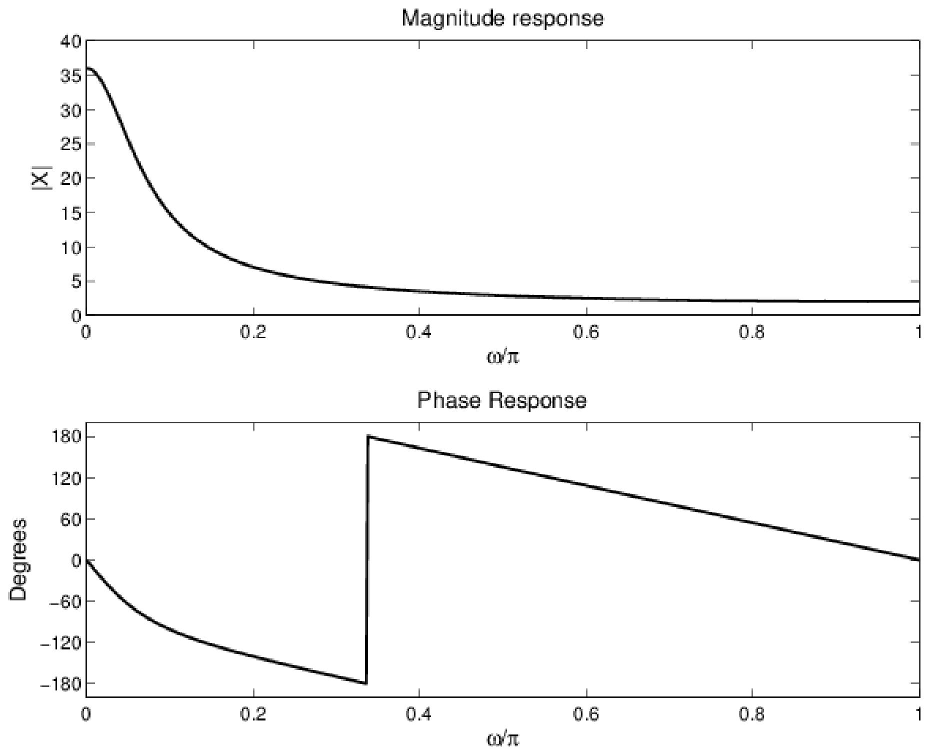


Figure 3.10: Problem P3.3.4 DTFT plots

5. $x(n) = 4\,(-0.7)^n \cos(0.25\pi n)u(n)$.

$$X(e^{j\omega}) = \sum_{-\infty}^{\infty} 4\,(-0.7)^n \cos(0.25\pi n)u(n)e^{-jn\omega} = 4\sum_{0}^{\infty} (-0.7)^n \cos(0.25\pi n)e^{-jn\omega}$$
$$= 4\frac{1-(-0.7)\cos(0.25\pi)e^{-j\omega}}{1-2(-0.7)e^{-j\omega}+(-0.7)^2e^{-j2\omega}} = 4\frac{1+0.495e^{-j\omega}}{1+1.4e^{-j\omega}+0.49e^{-j2\omega}}$$

Matlab Verification:

```
%% P0303e: DTFT of x5(n) = 4*((-0.7) ^
n)*cos(0.25*pi*n)*u(n)
clc; close all;
w5 = [0:500]*pi/500; X51 =
4*(ones(size(w5))+0.7*cos(0.25*pi)*exp(-1i*w5));
X52 = ones(size(w5))+1.4*cos(0.25*pi)*exp(-
1i*w5)+0.49*exp(-1i*2*w5);
X5 = X51./X52; magX5 = abs(X5); phaX5 = angle(X5);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0303e');
subplot(2,1,1); plot(w5/pi,magX5,'LineWidth',1.5);
axis([0 1 0 10]); wtick = [0:0.2:1]; magtick = [0:2:10];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w5/pi,phaX5*180/pi,'LineWidth',1.5);
axis([0,1,-200,200]); phatick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response','FontSize',12);
set(gca,'XTick',wtick);
set(gca,'YTick',phatick);
print -deps2 ../EPSFILES/P0303e;
```

The magnitude and phase plots of $X(e^{jw})$ are shown in Figure 3.11.

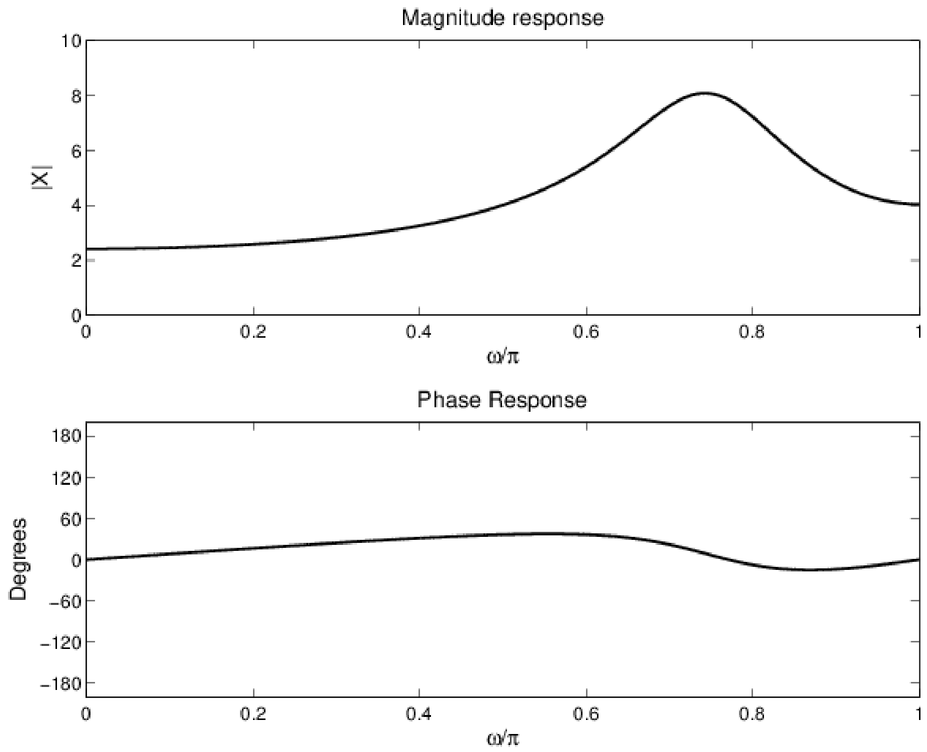


Figure 3.11: Problem P3.3.5 DTFT plots

P3.4

The following finite-duration sequences are called *windows* and are very useful in DSP.

$$\text{Rectangular: } \mathcal{R}_M(n) = \begin{cases} 1, & 0 \le n < M \\ 0, & \text{otherwise} \end{cases};$$

$$\text{Hanning: } \mathcal{C}_M(n) = 0.5\left[1 - \cos\frac{2\pi n}{M-1}\right]\mathcal{R}_M(n)$$

$$\text{Triangular: } \mathcal{T}_M(n) = \left[1 - \frac{|M-1-2n|}{M-1}\right]\mathcal{R}_M(n);$$

$$\text{Hamming: } \mathcal{H}_M(n) = \left[0.54 - 0.46\cos\frac{2\pi n}{M-1}\right]\mathcal{R}_M(n)$$

For each of these windows, determine their DTFTs for $M = 10, 25, 50, 101$. Scale transform values so that the maximum value is equal to 1. Plot the magnitude of the normalized DTFT over $-\pi \le \omega \le \pi$. Study these plots and comment on their behavior as a function of M.

Solutions

Window function DTFTs:

Rectangular Window: $\mathcal{R}_M(n) = u(n) - u(n-M)$

Matlab script:

```
% P3.4
```

```
%% P0304a: DTFT of a Rectangular Window, M = 10,25,50,101
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0304a');
w = linspace(-pi,pi,501); wtick = [-1:0.5:1]; magtick =
[0:0.5:1.1];
% M = 10
M = 10; n = 0:M; x = ones(1,length(n));
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
ylabel('|X|','FontSize',12); title(['M =
10'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 25
M = 25; n = 0:M; x = ones(1,length(n));
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,2); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
title(['M = 25'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 50
M = 50; n = 0:M; x = ones(1,length(n));
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,3); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['M = 50'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 101
M = 101; n = 0:M; x = ones(1,length(n));
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,4); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
xlabel('\omega/\pi','FontSize',12); title(['M =
101'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick); print -
deps2 ../EPSFILES/P0304a;
```

The magnitude plots of the DTFTs are shown in Figure 3.12.

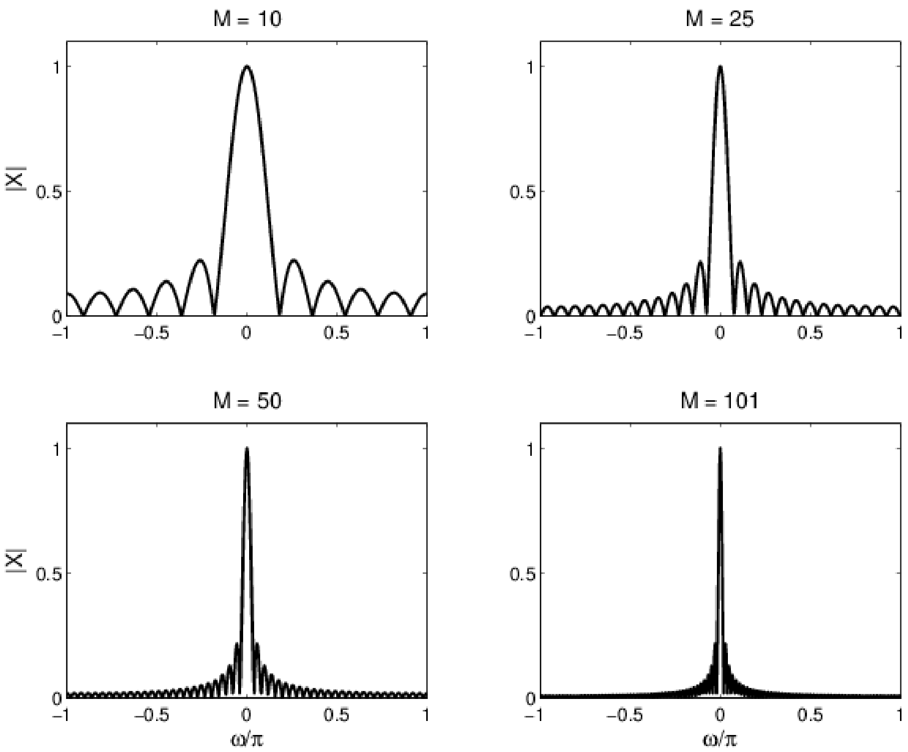


Figure 3.12: Problem P3.4 Rectangular window DTFT plots

Triangular Window: $\mathcal{T}_M(n) = \left[1 - \dfrac{|M-1-2n|}{M-1}\right]\mathcal{R}_M(n)$

Matlab script:

```
%% P0304b: DTFT of a Triangular Window,M = 10,25,50,101
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0304b');
w = linspace(-pi,pi,501); wtick = [-1:0.5:1]; magtick =
[0:0.5:1.1];
% M = 10
M = 10; n = 0:M; x = (1-(abs( M-1-(2*n) )/(M+1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
ylabel('|X|','FontSize',12); title(['M =
10'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 25
M = 25; n = 0:M; x = (1-(abs( M-1-(2*n) )/(M+1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,2); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
title(['M = 25'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
```

```
% M = 50
M = 50; n = 0:M; x = (1-(abs( M-1-(2*n) )/(M+1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,3); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['M = 50'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 101
M = 101; n = 0:M; x = (1-(abs( M-1-(2*n) )/(M+1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,4); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
xlabel('\omega/\pi','FontSize',12); title(['M =
101'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick); print -
deps2 ../EPSFILES/P0304b;
```

The magnitude plots of the DTFTs are shown in Figure 3.13.

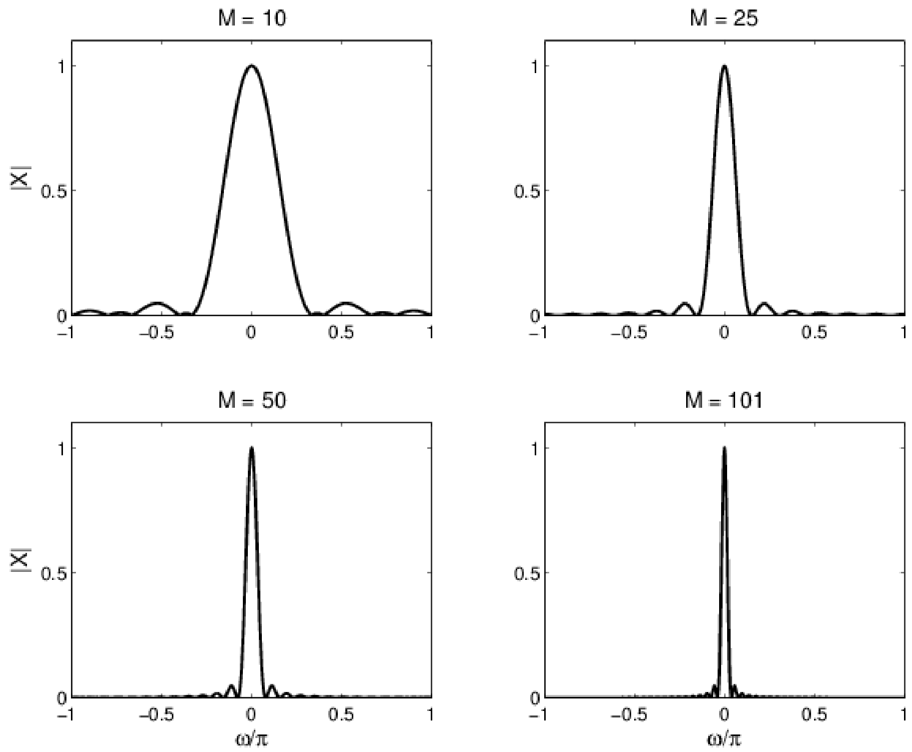


Figure 3.13: Problem P3.4 Triangular window DTFT plots

Hann Window: $\mathcal{C}_M(n) = 0.5\left[1 - \cos\dfrac{2\pi n}{M-1}\right]\mathcal{R}_M(n)$

Matlab script:

```
%% P0304c: DTFT of a Hann Window,M = 10,25,50,101
clc; close all;
```

```
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0304c');
w = linspace(-pi,pi,501); wtick = [-1:0.5:1]; magtick =
[0:0.5:1.1];
% M = 10
M = 10; n = 0:M; x = 0.5*(1-cos((2*pi*n)/(M-1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
ylabel('|X|','FontSize',12); title(['M =
10'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 25
M = 25; n = 0:M; x = 0.5*(1-cos((2*pi*n)/(M-1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,2); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
title(['M = 25'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 50
M = 50; n = 0:M; x = 0.5*(1-cos((2*pi*n)/(M-1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,3); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['M = 50'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 101
M = 101; n = 0:M; x = 0.5*(1-cos((2*pi*n)/(M-1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,4); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
xlabel('\omega/\pi','FontSize',12); title(['M =
101'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick); print -
deps2 ../EPSFILES/P0304c;
```
The magnitude plots of the DTFTs are shown in Figure 3.14.

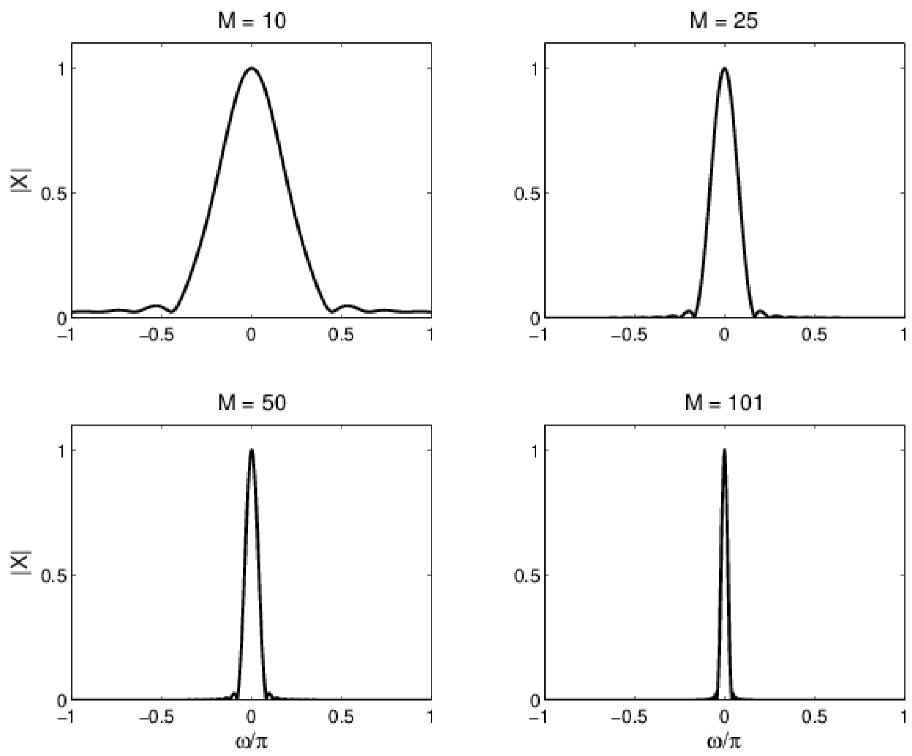


Figure 3.14: Problem P3.4 Hann window DTFT plots

Hamming Window: $\mathcal{H}_M(n) = \left[0.54 - 0.46\cos\frac{2\pi n}{M-1}\right]\mathcal{R}_M(n)$

Matlab script:

```
%% P0304d: DTFT of a Hamming Window,M = 10,25,50,101
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0304d');
w = linspace(-pi,pi,501); wtick = [-1:0.5:1]; magtick =
[0:0.5:1.1];
% M = 10
M = 10; n = 0:M; x = (0.54-0.46*cos((2*pi*n)/(M-1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
ylabel('|X|','FontSize',12); title(['M =
10'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 25
M = 25; n = 0:M; x = (0.54-0.46*cos((2*pi*n)/(M-1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,2); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
title(['M = 25'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
```

```
% M = 50
M = 50; n = 0:M; x = (0.54-0.46*cos((2*pi*n)/(M-1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,3); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['M = 50'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
% M = 101
M = 101; n = 0:M; x = (0.54-0.46*cos((2*pi*n)/(M-1)) );
X = dtft(x,n,w); magX = abs(X); magX = magX/max(magX);
subplot(2,2,4); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.1]);
xlabel('\omega/\pi','FontSize',12); title(['M =
101'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick); print -
deps2 ../EPSFILES/P0304d;
```

The magnitude plots of the DTFTs are shown in Figure 3.15.

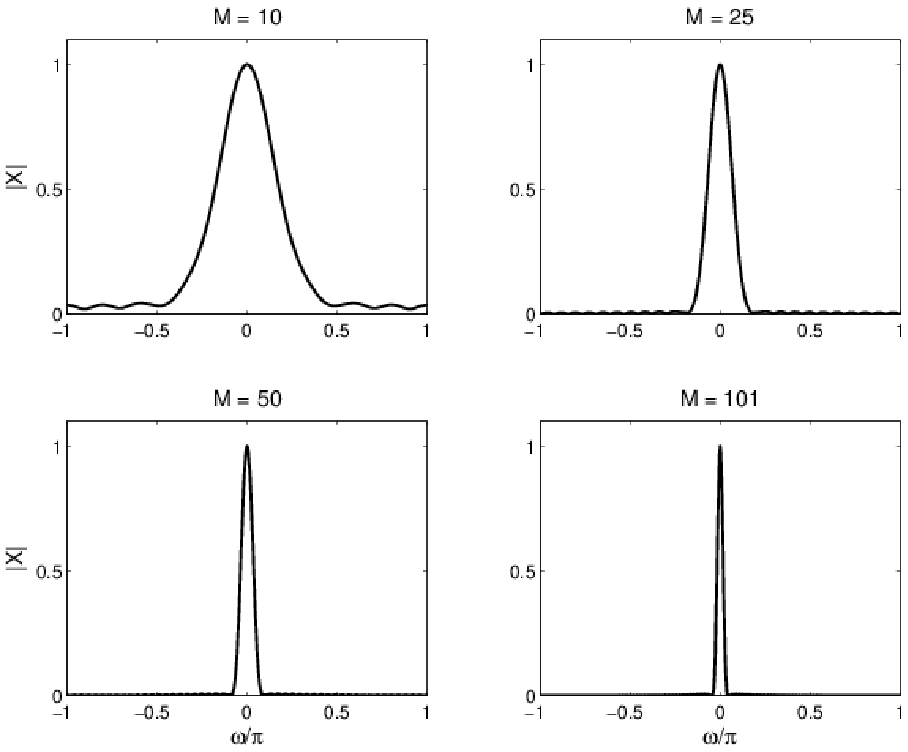


Figure 3.15: Problem P3.4 Hamming window DTFT plots

P3.5

Using the definition of the DTFT in (3.1), determine the sequences corresponding to the following DTFTs:

1. $X(e^{j\omega}) = 3 + 2\cos(\omega) + 4\cos(2\omega)$.
2. $X(e^{j\omega}) = [1 - 6\cos(3\omega) + 8\cos(5\omega)]\, e^{-j3\omega}$.
3. $X(e^{j\omega}) = 2 + j4\sin(2\omega) - 5\cos(4\omega)$.
4. $X(e^{j\omega}) = [1 + 2\cos(\omega) + 3\cos(2\omega)]\cos(\omega/2)e^{-j5\omega/2}$.
5. $X(e^{j\omega}) = j\,[3 + 2\cos(\omega) + 4\cos(2\omega)]\sin(\omega)e^{-j3\omega}$.

Solutions

Inverse DTFTs using the definition of the DTFT:

1. $X\left(e^{j\omega}\right) = 3 + 2\cos(\omega) + 4\cos(2\omega)$: Using the Euler identity

$$X\left(e^{j\omega}\right) = 3 + 2\frac{e^{j\omega} + e^{-j\omega}}{2} + 4\frac{e^{j2\omega} + e^{-j2\omega}}{2} = 2e^{j2\omega} + e^{j\omega} + 3 + e^{-j\omega} + 2e^{-j2\omega}$$

Hence $x(n) = \{2, 1, \underset{\uparrow}{3}, 1, 2\}$.

2. $X\left(e^{j\omega}\right) = [1 - 6\cos(3\omega) + 8\cos(5\omega)]\, e^{-j3\omega}$: Using the Euler identity

$$\begin{aligned} X\left(e^{j\omega}\right) &= \left[1 - 6\frac{e^{j3\omega} + e^{-j3\omega}}{2} + 8\frac{e^{j5\omega} + e^{-j5\omega}}{2}\right] e^{-j3\omega} \\ &= 4e^{j2\omega} - 3 + e^{-j3\omega} - 3e^{-j6\omega} + 4e^{-j8\omega} \end{aligned}$$

Hence $x(n) = \{4, 0, \underset{\uparrow}{-3}, 0, 0, 1, 0, 0, -3, 0, 4\}$.

3. $X\left(e^{j\omega}\right) = 2 + j4\sin(2\omega) - 5\cos(4\omega)$: Using the Euler identity

$$X\left(e^{j\omega}\right) = 2 + j4\frac{e^{j2\omega} - e^{-j2\omega}}{2j} - 5\frac{e^{j4\omega} + e^{-j4\omega}}{2} = -\tfrac{5}{2}e^{j4\omega} + 2e^{j2\omega} + 2 - 2e^{-j2\omega} - \tfrac{5}{2}e^{-j4\omega}$$

Hence $x(n) = \{-\tfrac{5}{2}, 0, 2, 0, \underset{\uparrow}{2}, 0, -2, 0, -\tfrac{5}{2}\}$.

4. $X\left(e^{j\omega}\right) = [1 + 2\cos(\omega) + 3\cos(2\omega)]\cos(\omega/2)e^{-j5\omega/2}$: Using the Euler identity

$$\begin{aligned} X\left(e^{j\omega}\right) &= \left[1 + 2\frac{e^{j\omega} + e^{-j\omega}}{2} + 3\frac{e^{j2\omega} + e^{-j2\omega}}{2}\right] \frac{e^{j\frac{1}{2}\omega} + e^{-j\frac{1}{2}\omega}}{2} e^{-j5\omega/2} \\ &= \left[\tfrac{3}{2}e^{j2\omega} + e^{j\omega} + 1 + e^{-j\omega} + \tfrac{3}{2}e^{-j2\omega}\right] \frac{e^{-j2\omega} + e^{-j3\omega}}{2} \\ &= \tfrac{3}{4} + \tfrac{5}{4}e^{-j\omega} + e^{-j2\omega} + e^{-j3\omega} + \tfrac{5}{4}e^{-j4\omega} + \tfrac{3}{4}e^{-j5\omega} \end{aligned}$$

Hence $x(n) = \{\underset{\uparrow}{\tfrac{3}{4}}, \tfrac{5}{4}, 1, 1, \tfrac{5}{4}, \tfrac{3}{4}\}$.

5. $X\left(e^{j\omega}\right) = j\,[3 + 2\cos(\omega) + 4\cos(2\omega)]\sin(\omega)e^{-j3\omega}$: Using the Euler identity

$$\begin{aligned} X\left(e^{j\omega}\right) &= j\left[3 + 2\frac{e^{j\omega} + e^{-j\omega}}{2} + 4\frac{e^{j2\omega} + e^{-j2\omega}}{2}\right] \frac{e^{j\omega} - e^{-j\omega}}{2j} e^{-j3\omega} \\ &= 2 + e^{-j\omega} + e^{-j2\omega} - e^{-j4\omega} - e^{-j5\omega} - 2e^{-j6\omega} \end{aligned}$$

Hence $x(n) = \{\underset{\uparrow}{2}, 1, 1, 0, -1, -1, -2\}$.

P3.6

Using the definition of the inverse DTFT in (3.2), determine the sequences corresponding to the following DTFTs:

1. $X(e^{j\omega}) = \begin{cases} 1, & 0 \le |\omega| \le \pi/3; \\ 0, & \pi/3 < |\omega| \le \pi. \end{cases}$
2. $X(e^{j\omega}) = \begin{cases} 0, & 0 \le |\omega| \le 3\pi/4; \\ 1, & 3\pi/4 < |\omega| \le \pi. \end{cases}$
3. $X(e^{j\omega}) = \begin{cases} 2, & 0 \le |\omega| \le \pi/8; \\ 1, & \pi/8 < |\omega| \le 3\pi/4. \\ 0, & 3\pi/4 < |\omega| \le \pi. \end{cases}$
4. $X(e^{j\omega}) = \begin{cases} 0, & -\pi \le |\omega| < \pi/4; \\ 1, & \pi/4 \le |\omega| \le 3\pi/4. \\ 0, & 3\pi/4 < |\omega| \le \pi. \end{cases}$
5. $X(e^{j\omega}) = \omega\, e^{j(\pi/2 - 10\omega)}$.

Remember that the above transforms are periodic in ω with period equal to 2π. Hence, functions are given only over the primary period of $-\pi \le \omega \le \pi$.

Solutions

Inverse DTFTs using the definition of the IDTFT:

1. $X(e^{j\omega}) = \begin{cases} 1, & 0 \le |\omega| \le \pi/3; \\ 0, & \pi/3 < |\omega| \le \pi. \end{cases}$

   Solution: Consider

$$x(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\, e^{jn\omega}\, d\omega = \frac{1}{2\pi}\int_{-\pi/3}^{\pi/3} e^{jn\omega}\, d\omega = \left.\frac{e^{jn\omega}}{j2\pi n}\right|_{-\pi/3}^{\pi/3} = \frac{\sin\left(\frac{\pi n}{3}\right)}{\pi n} = \frac{1}{3}\mathrm{sinc}\left(\frac{n}{3}\right)$$

2. $X(e^{j\omega}) = \begin{cases} 0, & 0 \le |\omega| \le 3\pi/4; \\ 1, & 3\pi/4 < |\omega| \le \pi. \end{cases}$

   Solution: Consider

$$\begin{aligned} x(n) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} X(e^{j\omega})\, e^{jn\omega}\, d\omega = \frac{1}{2\pi}\int_{-\pi}^{-3\pi/4} e^{jn\omega}\, d\omega + \frac{1}{2\pi}\int_{3\pi/4}^{\pi} e^{jn\omega}\, d\omega \\ &= \frac{2}{2\pi}\int_{3\pi/4}^{\pi} \cos(n\omega)\, d\omega = \frac{1}{\pi}\left.\frac{\sin(n\omega)}{n}\right|_{3\pi/4}^{\pi} = \delta(n) - \frac{3}{4}\mathrm{sinc}\left(\frac{3n}{4}\right) \end{aligned}$$

3. $X(e^{j\omega}) = \begin{cases} 2, & 0 \le |\omega| \le \pi/8; \\ 1, & \pi/8 < |\omega| \le 3\pi/4. \\ 0, & 3\pi/4 < |\omega| \le \pi. \end{cases}$

   Solution: Consider

$$\begin{aligned} x(n) &= \frac{2}{2\pi}\left[\int_{0}^{\pi/8} 2\cos(n\omega)\, d\omega + \int_{\pi/8}^{3\pi/4} \cos(n\omega)\, d\omega\right] = \frac{1}{\pi}\left[2\left.\frac{\sin(n\omega)}{n}\right|_{0}^{\pi/8} + \left.\frac{\sin(n\omega)}{n}\right|_{\pi/8}^{3\pi/4}\right] \\ &= \frac{1}{n\pi}\left[2\sin\left(\frac{n\pi}{8}\right) + \sin\left(\frac{3n\pi}{4}\right) - \sin\left(\frac{n\pi}{8}\right)\right] = \frac{1}{n\pi}\left[\sin\left(\frac{n\pi}{8}\right) + \sin\left(\frac{3n\pi}{4}\right)\right] \\ &= \frac{1}{8}\mathrm{sinc}\left(\frac{n}{8}\right) + \frac{3}{4}\mathrm{sinc}\left(\frac{3n}{4}\right) \end{aligned}$$

4. $X\left(e^{j\omega}\right)=\begin{cases}0, & -\pi \le \omega < \pi/4; \\ 1, & \pi/4 \le |\omega| \le 3\pi/4. \\ 0, & 3\pi/4 < |\omega| \le \pi.\end{cases}$

Solution: Consider

$$x(n)=\frac{2}{2\pi}\int_{\pi/4}^{3\pi/4}\cos(n\omega)\,d\omega=\frac{\sin(n\omega)}{n\pi}\bigg|_{\pi/4}^{3\pi/4}=\frac{\sin\left(\frac{3n\pi}{4}\right)-\sin\left(\frac{n\pi}{4}\right)}{n\pi}=\frac{3}{4}\text{sinc}\left(\frac{3n}{4}\right)-\frac{1}{4}\text{sinc}\left(\frac{n}{4}\right)$$

5. $X\left(e^{j\omega}\right)=\omega\, e^{j(\pi/2-10\omega)}$.

Solution: Consider

$$x(n)=\frac{1}{2\pi}\int_{-\pi}^{\pi}\omega\, e^{j(\pi/2-10\omega)}e^{jn\omega}d\omega=\frac{j}{2\pi}\int_{-\pi}^{\pi}\omega\, e^{j(n-10)\omega}d\omega$$
$$=\frac{j}{2\pi}\left[\frac{\omega e^{j(n-10)\omega}}{j(n-10)}\bigg|_{-\pi}^{\pi}-\int_{-\pi}^{\pi}\frac{e^{j(n-10)\omega}}{j(n-10)}\right]=\cos[(n-10)\pi]-\frac{\sin[(n-10)\pi]}{\pi(n-10)^2}$$

P3.7

A complex-valued sequence $x(n)$ can be decomposed into a conjugate symmetric part $xe(n)$ and an conjugate anti-symmetric part $xo(n)$ as discussed in Chapter 2. Show that

$$\mathcal{F}\left[x_e(n)\right]=X_R(e^{j\omega}) \quad \text{and} \quad \mathcal{F}\left[x_o(n)\right]=jX_I(e^{j\omega})$$

where $X_R(e^{jw})$ and $X_I(e^{jw})$ are the real and imaginary parts of the DTFT $X(e^{jw})$ respectively. Verify this property on

$$x(n) = 2(0.9)^{-n}\ [\cos(0.1\pi n) + j\sin(0.9\pi n)]\ [u(n) - u(n - 10)]$$

using the MATLAB functions developed in Chapter 2

Solutions

A complex-valued sequence $x(n)$ can be decomposed into a conjugate symmetric part $x_e(n)$ and an conjugate anti-symmetric part $x_o(n)$ as

$$x_e(n)=\frac{1}{2}\left[x(n)+x^*(-n)\right]; \qquad x_o(n)=\frac{1}{2}\left[x(n)-x^*(-n)\right]$$

Consider

$$\mathcal{F}\left[x_e(n)\right]=\sum_{-\infty}^{\infty}x_e(n)e^{-jn\omega}=\sum_{-\infty}^{\infty}\frac{1}{2}\left[x(n)+x^*(-n)\right]e^{-jn\omega}=\frac{1}{2}\left[\sum_{-\infty}^{\infty}x(n)e^{-jn\omega}+\sum_{-\infty}^{\infty}x^*(-n)e^{-jn\omega}\right]$$
$$=\frac{1}{2}\left[X\left(e^{j\omega}\right)+X^*\left(e^{j\omega}\right)\right]=X_R\left(e^{j\omega}\right)$$

Similarly,

$$\mathcal{F}\left[x_o(n)\right]=\sum_{-\infty}^{\infty}x_o(n)e^{-jn\omega}=\sum_{-\infty}^{\infty}\frac{1}{2}\left[x(n)-x^*(-n)\right]e^{-jn\omega}=\frac{1}{2}\left[\sum_{-\infty}^{\infty}x(n)e^{-jn\omega}-\sum_{-\infty}^{\infty}x^*(-n)e^{-jn\omega}\right]$$
$$=\frac{1}{2}\left[X\left(e^{j\omega}\right)-X^*\left(e^{j\omega}\right)\right]=jX_I\left(e^{j\omega}\right)$$

Matlab Verification using $x(n) = 2(0.9)^{-n}$ [cos(0.1πn) + j sin(0.9πn)] [u(n) - u(n - 10)]:

```
% P3.7
%% P0307: DTFT after even and odd part decomposition of
x(n)
```

```
% x(n) = 2*(0.9)^(-
n)*(cos(0.1*pi*n)+j*sin(0.9*pi*n))(u(n)-u(n-10))
clc; close all;
%
[x1,n1] = stepseq(0,0,10); [x2,n2] = stepseq(10,0,10);
[x3,n3] = sigadd(x1,n1,-x2,n2);
n = n3; x = 2*(0.9 .^ (-n)).*(cos(0.1*pi*n)+1i*sin(0.9*pi
*n)).*x3;
[xe,xo,m] = evenodd(x,n);
w = [-500:500]*pi/500; X = dtft(x,n,w); realX = real(X);
imagX = imag(X);
Xe = dtft(xe,m,w); Xo = dtft(xo,m,w);
diff_e = max(abs(realX-Xe)); diff_o = max(abs(1i*imagX-
Xo));
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0307');
subplot(2,2,1); plot(w/pi,real(Xe),'LineWidth',1.5);
axis([-1 1 -30 20]); wtick = sort([-1:0.4:1 0]); magtick
= [-30:10:20];
xlabel('\omega/\pi','FontSize',12);
ylabel('X_e','FontSize',12);
title('DTFT of even part of x(n)','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,3); plot(w/pi,realX,'LineWidth',1.5); axis([-
1 1 -30 20]);
wtick = sort([-1:0.4:1 0] ); magtick = [-30:10:20];
xlabel('\omega/\pi','FontSize',12);
ylabel('X_R','FontSize',12);
title('Real part:DTFT of x(n)','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,2); plot(w/pi,imag(Xo),'LineWidth',1.5);
axis([-1 1 -30 20]);
wtick = sort([-1:0.4:1 0]); magtick = [-30:10:20];
xlabel('\omega/\pi','FontSize',12);
ylabel('X_o','FontSize',12);
title('DTFT of odd part of x(n)','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,4); plot(w/pi,imagX,'LineWidth',1.5); axis([-
1 1 -30 20]);
wtick = sort([-1:0.4:1 0]); magtick = [-30:10:20];
xlabel('\omega/\pi','FontSize',12);
ylabel('X_I','FontSize',12);
title('Imaginary part:DTFT of x(n)','FontSize',12);
```

```
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0307;
```

The magnitude plots of the DTFTs are shown in Figure 3.16.

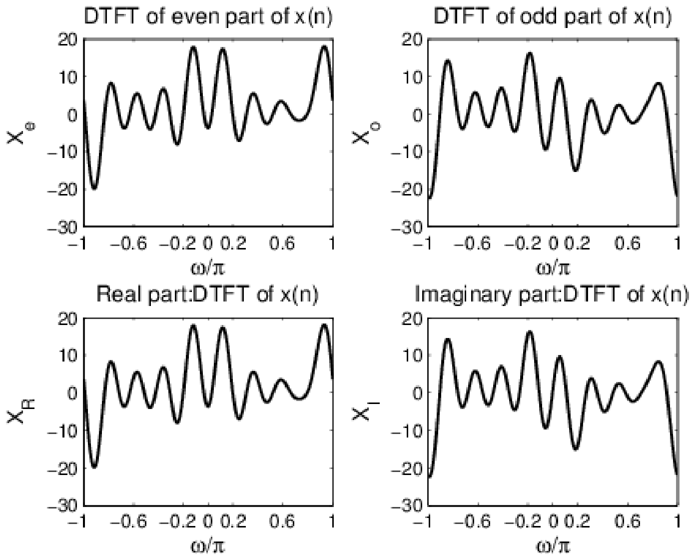


Figure 3.16: Problem P3.7 DTFT plots

P3.8

A complex-valued DTFT $X(e^{jw})$ can also be decomposed into its conjugate symmetric part $X_e(e^{jw})$ and conjugate anti-symmetric part $X_o(e^{jw})$, i.e.,

$$X(e^{j\omega}) = X_e(e^{j\omega}) + X_o(e^{j\omega})$$

where

$$X_e(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega}) + X^*(e^{-j\omega})] \quad \text{and} \quad X_0(e^{j\omega}) = \frac{1}{2}[X(e^{j\omega}) - X^*(e^{-j\omega})]$$

Show that

$$\mathcal{F}^{-1}[X_e(e^{j\omega})] = x_R(n) \quad \text{and} \quad \mathcal{F}^{-1}[X_0(e^{j\omega})] = jx_I(n)$$

where $x_R(n)$ and $x_I(n)$ are the real and imaginary parts of $x(n)$. Verify this property on

$$x(n) = e^{j0.1\pi n}\,[u(n) - u\,(n-20)]$$

using the MATLAB functions developed in Chapter 2.

Solutions

A complex-valued DTFT $X\left(e^{jw}\right)$ can be decomposed into its conjugate symmetric part $X_e\left(e^{jw}\right)$ and conjugate anti-symmetric part $X_o\left(e^{jw}\right)$, as

$$X(e^{j\omega}) = X_e(e^{j\omega}) + X_o(e^{j\omega});\ X_e(e^{j\omega}) = \tfrac{1}{2}\left[X(e^{j\omega}) + X^*(e^{-j\omega})\right],\ X_0(e^{j\omega}) = \tfrac{1}{2}\left[X(e^{j\omega}) - X^*(e^{-j\omega})\right]$$

Consider

$$\mathcal{F}^{-1}\left[X_e(e^{j\omega})\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_e(e^{j\omega})\, e^{jn\omega}\, d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} \tfrac{1}{2}\left[X(e^{j\omega}) + X^*(e^{-j\omega})\right] e^{jn\omega}\, d\omega$$
$$= \tfrac{1}{2}[x(n) + x^*(-n)] = x_R(n)$$

Similarly,

$$\mathcal{F}^{-1}\left[X_o(e^{j\omega})\right] = \frac{1}{2\pi}\int_{-\pi}^{\pi} X_o(e^{j\omega})\, e^{jn\omega}\, d\omega = \frac{1}{2\pi}\int_{-\pi}^{\pi} \tfrac{1}{2}\left[X(e^{j\omega}) - X^*(e^{-j\omega})\right] e^{jn\omega}\, d\omega$$
$$= \tfrac{1}{2}[x(n) - x^*(-n)] = jx_I(n)$$

Matlab Verification using $x(n) = e^{j0.1\pi n}\,[u(n) - u(n - 20)]$:

```
% P3.8
%% P0308: x(n) = exp(0.1*j*pi*n)*(u(n)-u(n-20));
clc; close all;
% set(0,'defaultfigurepaperposition',[0,0,6,6]);
%
[x1,n1] = stepseq(0,0,20); [x2,n2] = stepseq(20,0,20);
[x3,n3] = sigadd(x1,n1,-x2,n2); n = n3; x =
exp(0.1*1i*pi*n).*x3;
w1 = [-500:500]*pi/500; X = dtft(x,n,w1); [Xe,Xo,w2] =
evenodd(X,[-500:500]);
w2 = w2*pi/500; xr = real(x); xi = imag(x); Xr =
dtft(xr,n,w1);
Xi = dtft(1i*xi,n,w1); diff_r = max(abs(Xr-Xe)); diff_i =
max(abs(Xi-Xo));
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0308');
subplot(4,2,1); plot(w1/pi,abs(Xr),'LineWidth',1.5);
ylabel('|X_r|','FontSize',12);
title('Magnitude response of x_R','FontSize',12);
subplot(4,2,2);
plot(w1/pi,angle(Xr)*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]); magtick = [-180:90:180];
ylabel('Degrees','FontSize',12);
title('Phase response of x_R','FontSize',12);
set(gca,'YTick',magtick);
subplot(4,2,3); plot(w1/pi,abs(Xe),'LineWidth',1.5);
axis([-1 1 0 15]);
ytick = [0:5:15]; ylabel('|X_e|','FontSize',12);
title([ 'Magnitude part of X_e' ],'FontSize',12);
set(gca,'YTick',ytick);
subplot(4,2,4);
plot(w1/pi,angle(Xe)*180/pi,'LineWidth',1.5);
```

```
axis([-1 1 -200 200]); magtick = [-180:90:180];
ylabel('Degrees','FontSize',12);
title([ 'Phase part of X_e' ],'FontSize',12);
set(gca,'YTick',magtick);
subplot(4,2,5); plot(w1/pi,abs(Xi),'LineWidth',1.5);
ytick = [0:5:15]; axis([-1 1 0 15]);
ylabel('|X_i|','FontSize',12);
title([ 'Magnitude response of j*x_I' ],'FontSize',12);
set(gca,'YTick',ytick);
subplot(4,2,6);
plot(w1/pi,angle(Xi)*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]); magtick = [-180:90:180];
ylabel('Degrees','FontSize',12);
title([ 'Phase response of j*x_I' ],'FontSize',12);
set(gca,'YTick',magtick);
subplot(4,2,7); plot(w1/pi,abs(Xo),'LineWidth',1.5);
ytick = [0:5:15]; axis([-1 1 0 15]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X_o|','FontSize',12);
title(['Magnitude part of X_o'],'FontSize',12);
set(gca,'YTick',ytick);
subplot(4,2,8);
plot(w1/pi,angle(Xo)*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]); magtick = [-180:90:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase part of X_o'],'FontSize',12);
set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto'); print -
deps2 ../EPSFILES/P0308;
```

The magnitude plots of the DTFTs are shown in Figure 3.17.

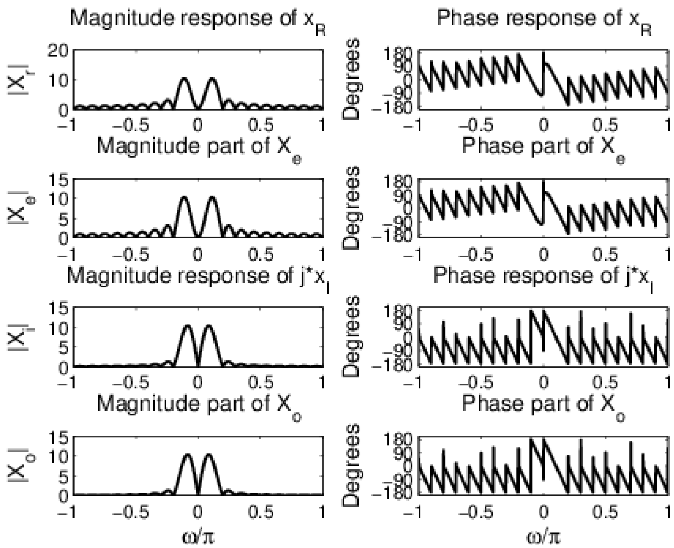


Figure 3.17: Problem P3.8 DTFT plots

P3.9

Using the frequency-shifting property of the DTFT, show that the real part of $X(e^{jw})$ of a sinusoidal pulse

$$x(n) = (\cos \omega_o n)\mathcal{R}_M(n)$$

where $R_M(n)$ is the rectangular pulse given in Problem P3.4 is given by

$$X_R(e^{j\omega}) = \frac{1}{2}\cos\left\{\frac{(\omega-\omega_0)(M-1)}{2}\right\}\frac{\sin\{(\omega-\omega_0)M/2\}}{\sin\{(\omega-\omega_0)/2\}} + \frac{1}{2}\cos\left\{\frac{(\omega+\omega_0)(M-1)}{2}\right\}\frac{\sin\{[\omega-(2\pi-\omega_0)]M/2\}}{\sin\{[\omega-(2\pi-\omega_0)]/2\}}$$

Compute and plot $X_R(e^{jw})$ for $\omega_o = \pi/2$ and M = 5, 15, 25, 100. Use the plotting interval $[-\pi, \pi]$. Comment on your results.

Solutions

The real-part of the DTFT of a sinusoidal pulse $x(n) = (\cos\omega_0 n)\ R_M(n)$:
First note that if the sequence $x(n)$ is a real-valued sequence, then the real part of its DTFT $X(e^{jw})$ is given by

$$X_R(e^{j\omega}) = \sum_{-\infty}^{\infty} x(n)\cos(n\omega)$$

Hence for the given sinusoidal pulse, we have

$$X_R(e^{j\omega}) = \sum_{0}^{M-1}\cos(\omega_0 n)\cos(n\omega) = \frac{1}{2}\sum_{0}^{M-1}\cos[(\omega-\omega_0)n] + \frac{1}{2}\sum_{0}^{M-1}\cos[(\omega+\omega_0)n] \qquad (3.2)$$

Consider the first sum in (3.2),

$$\sum_0^{M-1}\cos[(\omega-\omega_0)n]=\frac{1}{2}\sum_0^{M-1}\left\{e^{j(\omega-\omega_0)n}+e^{-j(\omega-\omega_0)n}\right\}=\frac{1}{2}\left\{\frac{1-e^{j(\omega-\omega_0)M}}{1-e^{j(\omega-\omega_0)}}+\frac{1-e^{-j(\omega-\omega_0)M}}{1-e^{-j(\omega-\omega_0)}}\right\}$$
$$=\frac{1}{2}\left(\frac{1-\cos(\omega-\omega_0)-\cos[(\omega-\omega_0)M]+\cos[(\omega-\omega_0)(M-1)]}{1-\cos(\omega-\omega_0)}\right)$$
$$=\frac{1}{2}\left(\frac{2\sin^2[(\omega-\omega_0)/2]+2\sin[(\omega-\omega_0)/2]\sin[(\omega-\omega_0)(M-\frac{1}{2})]}{2\sin^2[(\omega-\omega_0)/2]}\right)$$
$$=\frac{1}{2}\left(\frac{\sin[(\omega-\omega_0)/2]+\sin[(\omega-\omega_0)(M-\frac{1}{2})]}{\sin[(\omega-\omega_0)/2]}\right)$$
$$=\frac{\cos[(\omega-\omega_0)(M-1)]\sin[(\omega-\omega_0)M/2]}{\sin[(\omega-\omega_0)/2]} \tag{3.3}$$

Similarly,

$$\sum_0^{M-1}\cos[(\omega+\omega_0)n]=\sum_0^{M-1}\cos[\{\omega-(2\pi-\omega_0)\}n]$$
$$=\frac{\cos[\{\omega-(2\pi-\omega_0)\}(M-1)]\sin[\{\omega-(2\pi-\omega_0)\}M/2]}{\sin[\{\omega-(2\pi-\omega_0)\}/2]} \tag{3.4}$$

Substituting (3.3) and (3.4) in (3.2), we obtain the desired result.

Matlab Computation and plot of $X_R(e^{jw})$ for $\boldsymbol{\omega_0}$ = π/2 and M = 5, 15, 25, 100:

```
% P3.9
% P0309: DTFT of sinusoidal pulse for different values of
M
clc; close all;
%% M = 5
M = 5; w0 = pi/2; w = [-500:500]*pi/500;
X = 0.5*(exp(-1i*(w-w0)*((M-1)/2)).*sin((w-
w0)*M/2)./sin((w-w0+eps)/2)) + ...
0.5* (exp(-1i*(w+w0)*((M-
1)/2)).*sin((w+w0)*M/2)./sin((w+w0+eps)/2));
magX = abs(X); phaX = angle(X);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0309');
subplot(4,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 4]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Magnitude Response M = 5'],'FontSize',12);
subplot(4,2,2); plot(w/pi,phaX*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]);
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase Response M = 5'],'FontSize',12);
ytick = [-180 0 180];
set(gca,'YTickmode','manual','YTick',ytick);
```

```
%% M = 15
M = 15; w0 = pi/2; w = [-500:500]*pi/500;
X = 0.5*(exp(-1i*(w-w0)*((M-1)/2)).*sin((w-
w0)*M/2)./sin((w-w0+eps)/2)) + ...
0.5* (exp(-1i*(w+w0)*((M-
1)/2)).*sin((w+w0)*M/2)./sin((w+w0+eps)/2));
magX = abs(X); phaX = angle(X);
%
subplot(4,2,3); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 10]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title([char(10) 'Magnitude Response M = 15'
char(10)],'FontSize',12);
subplot(4,2,4); plot(w/pi,phaX*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]);
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title([char(10) 'Phase Response M = 15'
char(10)],'FontSize',12);
ytick = [-180 0 180];
set(gca,'YTickmode','manual','YTick',ytick);
%% M = 25
M = 25; w0 = pi/2; w = [-500:500]*pi/500;
X = 0.5*(exp(-1i*(w-w0)*((M-1)/2)).*sin((w-
w0)*M/2)./sin((w-w0+eps)/2)) + ...
0.5* (exp(-1i*(w+w0)*((M-
1)/2)).*sin((w+w0)*M/2)./sin((w+w0+eps)/2));
magX = abs(X); phaX = angle(X);
%
subplot(4,2,5); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 15]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title([char(10) 'Magnitude Response M = 25'
char(10)],'FontSize',12);
subplot(4,2,6); plot(w/pi,phaX*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]);
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title([char(10) 'Phase Response M = 25'
char(10)],'FontSize',12);
ytick = [-180 0
180];set(gca,'YTickmode','manual','YTick',ytick);
```

```
%%M = 101
M = 101; w0 = pi/2; w = [-500:500]*pi/500;
X = 0.5*(exp(-1i*(w-w0)*((M-1)/2)).*sin((w-
w0)*M/2)./sin((w-w0+eps)/2)) + ...
0.5* (exp(-1i*(w+w0)*((M-
1)/2)).*sin((w+w0)*M/2)./sin((w+w0+eps)/2));
magX = abs(X); phaX = angle(X);
subplot(4,2,7); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 75]);
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title([char(10) 'Magnitude Response M = 101'
char(10)],'FontSize',12);
ytick = [0 50
75];set(gca,'YTickmode','manual','YTick',ytick);
subplot(4,2,8); plot(w/pi,phaX*180/pi,'LineWidth',1.5);
axis([-1 1 -200 200]);
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title([char(10) 'Phase Response M = 101'
char(10)],'FontSize',12);
ytick = [-180 0
180];set(gca,'YTickmode','manual','YTick',ytick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0309;
```

The plots of $X_R(e^{jw})$ are shown in Figure 3.18.

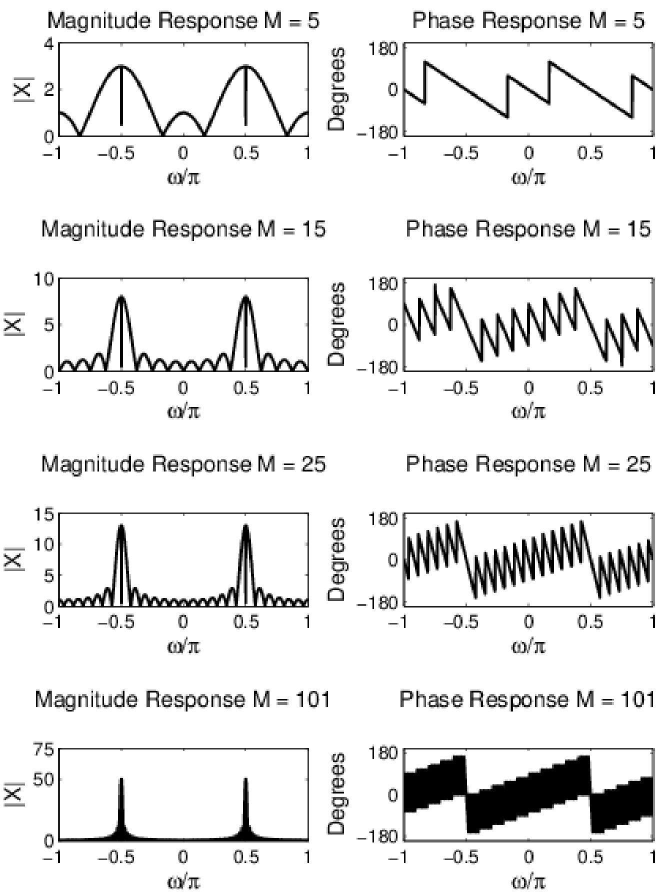


Figure 3.18: Problem P3.9 plots

P3.10

Let $x(n) = T_{10}(n)$ be a triangular pulse given in Problem P3.4. Using properties of the DTFT, determine and plot the DTFT of the following sequences.

1. $x(n) = \mathcal{T}_{10}(-n)$
2. $x(n) = \mathcal{T}_{10}(n) - \mathcal{T}_{10}(n-10)$
3. $x(n) = \mathcal{T}_{10}(n) * \mathcal{T}_{10}(-n)$
4. $x(n) = \mathcal{T}_{10}(n)e^{j\pi n}$
5. $x(n) = \cos(0.1\pi n)\mathcal{T}_{10}(n)$

Solutions

$x(n) = T_{10}(n)$ is a triangular pulse given in Problem P4.3. DTFT calculations and plots using properties of the DTFT:

1. $x(n) = T_{10}(-n)$:

$$\begin{aligned} X\left(e^{j\omega}\right) &= \mathcal{F}[x(n)] = \mathcal{F}[\mathcal{T}_{10}(-n)] = \mathcal{F}[\mathcal{T}_{10}(n)]|_{\omega\to-\omega} \\ &= \mathcal{F}[\mathcal{T}_{10}(n)]^* \text{ } (\because \text{real signal} \end{aligned}$$

Matlab script:

```
% P3.10
%% P0310a: DTFT of x(n) = T_10(-n)
clc; close all;
% Triangular Window T_10(n) & its DTFT
M = 10; n = 0:M; Tn = (1-(abs(M-1-2*n)/(M+1)));
w = linspace(-pi,pi,501); Tw = dtft(Tn,n,w); magTw =
abs(Tw);
magTw = magTw/max(magTw); phaTw = angle(Tw)*180/pi;
% x(n) & its DTFT
[x,n] = sigfold(Tn,n); X = dtft(x,n,w); magX = abs(X);
magX = magX/max(magX); phaX = angle(X)*180/pi;
% DTFT of x(n) from the Property
Y = fliplr(Tw); magY = abs(Y)/max(abs(Y)); phaY =
angle(Y)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0310a');
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,2); plot(w/pi,phaX,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.5:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,3); plot(w/pi,magY,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude from the
Property'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,4); plot(w/pi,phaY,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.4:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
```

```
ylabel('Degrees','FontSize',12);
title(['Phase from the Property'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0310a;
```

The property verification using plots of $X(e^{jw})$ is shown in Figure 3.19.

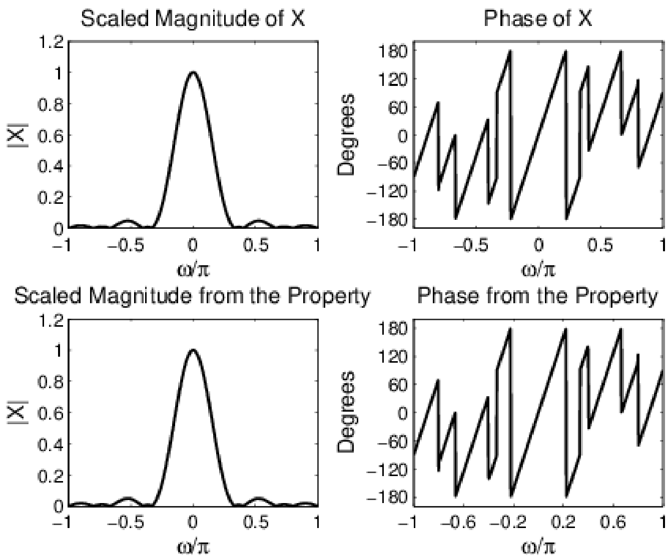


Figure 3.19: Problem P3.10a plots

2. $x(n) = \mathcal{T}_{10}(n) - \mathcal{T}_{10}(n-10)$:

$$\begin{aligned} X\left(e^{j\omega}\right) &= \mathcal{F}[x(n)] = \mathcal{F}[\mathcal{T}_{10}(n) - \mathcal{T}_{10}(n-10)] = \mathcal{F}[\mathcal{T}_{10}(n)] - \mathcal{F}[\mathcal{T}_{10}(n)]e^{-j10\omega} \\ &= \left(1 - e^{-j10\omega}\right)\mathcal{F}[\mathcal{T}_{10}(n)] \end{aligned}$$

Matlab script:

```
%% P0310b: DTFT of T_10(n)-T_10(n-10);
% % T_10(n) = [1-(abs(M-1-2*n)/(M+1)) ]*R_M(n), M = 10
clc; close all;
% Triangular Window T_10(n) & its DTFT
M = 10; n = 0:M; Tn = (1-(abs(M-1-2*n)/(M+1)));
w = linspace(-pi,pi,501); Tw = dtft(Tn,n,w); magTw =
abs(Tw);
magTw = magTw/max(magTw); phaTw = angle(Tw)*180/pi;
% x(n) & its DTFT
[x1,n1] = sigshift(Tn,n,10); [x,n] = sigadd(Tn,n,-x1,n1);
X = dtft(x,n,w); magX = abs(X);
magX = magX/max(magX); phaX = angle(X)*180/pi;
% DTFT of x(n) from the Property
Y = Tw-exp(-1i*w*10).*Tw; magY = abs(Y)/max(abs(Y)); phaY
= angle(Y)*180/pi;
%
```

```
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0310b');
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,2); plot(w/pi,phaX,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.5:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,3); plot(w/pi,magY,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude from the
Property'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,4); plot(w/pi,phaY,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.4:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase from the Property'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0310b;
```

The property verification using plots of $X\left(e^{jw}\right)$ is shown in Figure 3.20.

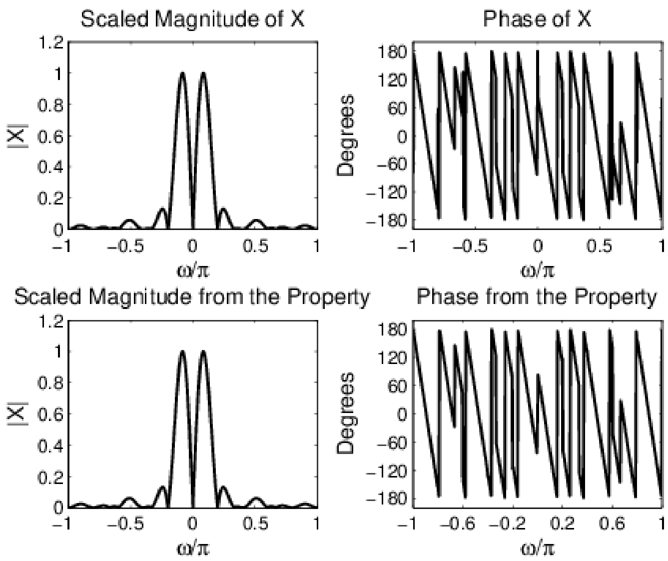


Figure 3.20: Problem P3.10b plots

3. $x(n) = T_{10}(n) * T_{10}(-n)$:

$$X\left(e^{j\omega}\right) = \mathcal{F}[x(n)] = \mathcal{F}[T_{10}(n) * T_{10}(-n)] = \mathcal{F}[T_{10}(n)] * \mathcal{F}[T_{10}(n)]^*$$
$$= |\mathcal{F}[T_{10}(n)]|^2$$

Matlab script:

```
%% P0310c: DTFT of T_10(n) Conv T_10(-n)
% T_10(n) = [1-(abs(M-1-2*n)/(M+1)) ]*R_M(n), M = 10
clc; close all;
% Triangular Window T_10(n) & its DTFT
M = 10; n = 0:M; Tn = (1-(abs(M-1-2*n)/(M+1)));
w = linspace(-pi,pi,501); Tw = dtft(Tn,n,w); magTw =
abs(Tw);
magTw = magTw/max(magTw); phaTw = angle(Tw)*180/pi;
% x(n) & its DTFT
[x1,n1] = sigfold(Tn,n); [x,n] = conv_m(Tn,n,x1,n1);
X = dtft(x,n,w); magX = abs(X);
magX = magX/max(magX); phaX = angle(X)*180/pi;
% DTFT of x(n) from the Property
Y = Tw.*fliplr(Tw); magY = abs(Y)/max(abs(Y)); phaY =
angle(Y)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0310c');
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude of X'],'FontSize',12);
```

```
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,2); plot(w/pi,phaX,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.5:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,3); plot(w/pi,magY,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude from the
Property'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,4); plot(w/pi,phaY,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.4:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase from the Property'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);set(gcf,'paperposi
tionmode','auto');
print -deps2 ../EPSFILES/P0310c;
```

The property verification using plots of $X(e^{jw})$ is shown in Figure 3.21.

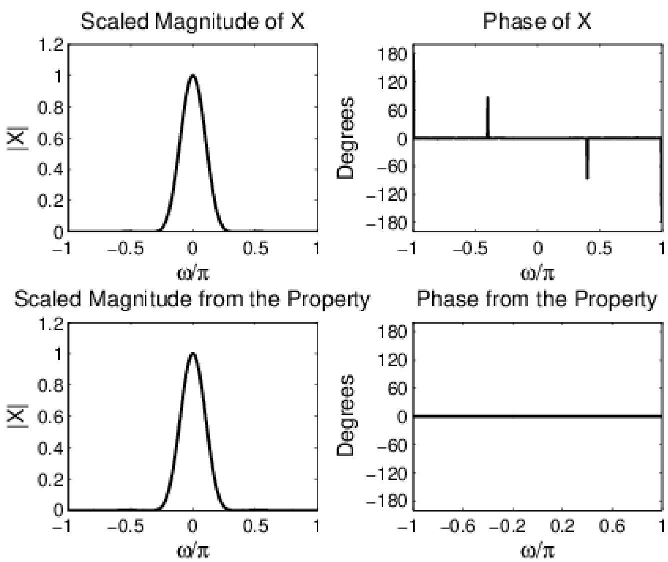


Figure 3.21: Problem P3.10c plots

4. $x(n) = T_{10}(n)e^{j\pi n}$:

$$X\left(e^{j\omega}\right) = \mathcal{F}[x(n)] = \mathcal{F}[\mathcal{T}_{10}(n)e^{j\pi n}]$$
$$= \mathcal{F}[\mathcal{T}_{10}(n)]|_{\omega\to(\omega-\pi)}$$

Matlab script:

```
%% P0310d: DTFT of T_10(n)*exp(1i*pi*n)
% T_10(n) = [1-(abs(M-1-2*n)/(M+1)) ]*R_M(n), M = 10
clc; close all;
% Triangular Window T_10(n) & its DTFT
M = 10; n = 0:M; Tn = (1-(abs(M-1-2*n)/(M+1)));
w = linspace(-pi,pi,501); Tw = dtft(Tn,n,w); magTw =
abs(Tw);
magTw = magTw/max(magTw); phaTw = angle(Tw)*180/pi;
% x(n) & its DTFT
x = Tn.*exp(1i*pi*n); X = dtft(x,n,w); magX = abs(X);
magX = magX/max(magX); phaX = angle(X)*180/pi;
% DTFT of x(n) from the Property
Y = [Tw(251:501),Tw(1:250)]; magY = abs(Y)/max(abs(Y));
phaY = angle(Y)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0310d');
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,2); plot(w/pi,phaX,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.5:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,3); plot(w/pi,magY,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude from the
Property'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,4); plot(w/pi,phaY,'LineWidth',1.5); axis([-1
```

```
1 -200 200]);
wtick = sort([-1:0.4:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase from the Property'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);set(gcf,'paperposi
tionmode','auto');
print -deps2 ../EPSFILES/P0310d;
```

The property verification using plots of $X(e^{jw})$ is shown in Figure 3.22.

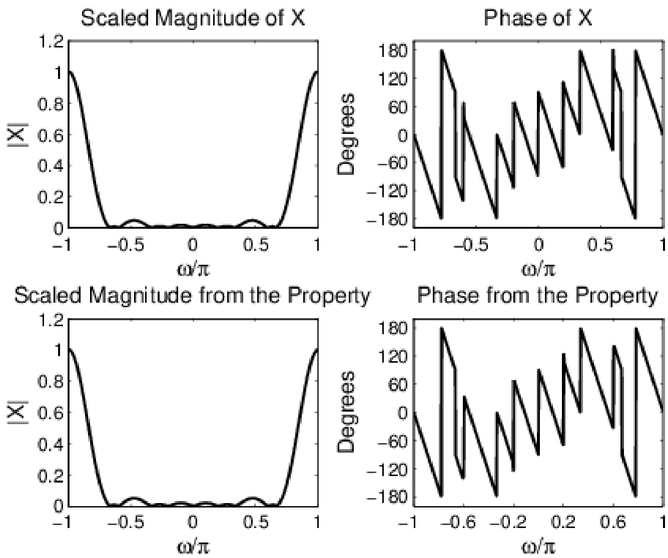


Figure 3.22: Problem P3.10d plots

5. $x(n) = \cos(0.1\pi n)\, T_{10}(n)$:

$$X\left(e^{j\omega}\right) = \mathcal{F}[x(n)] = \mathcal{F}[\cos(0.1\pi n)\mathcal{T}_{10}(n)] = \mathcal{F}\left[\frac{1}{2}\left\{e^{j0.1\pi n} + e^{-j0.1\pi n}\right\}\mathcal{T}_{10}(n)\right]$$

$$= \frac{1}{2}\,\mathcal{F}[\mathcal{T}_{10}(n)]|_{\omega\to(\omega-0.1\pi)} + \frac{1}{2}\,\mathcal{F}[\mathcal{T}_{10}(n)]|_{\omega\to(\omega+0.1\pi)}$$

Matlab script:

```
%% P0310e: DTFT of x(n) = T_10(-n)
clc; close all;
% Triangular Window T_10(n) & its DTFT
M = 10; n = 0:M; Tn = (1-(abs(M-1-2*n)/(M+1)));
w = linspace(-pi,pi,501); Tw = dtft(Tn,n,w); magTw =
abs(Tw);
magTw = magTw/max(magTw); phaTw = angle(Tw)*180/pi;
% x(n) & its DTFT
x = cos(0.1*pi*n).*Tn; X = dtft(x,n,w); magX = abs(X);
magX = magX/max(magX); phaX = angle(X)*180/pi;
% DTFT of x(n) from the Property
Y = 0.5*([Tw(477:501),Tw(1:476)]+[Tw(26:501),Tw(1:25)]);
```

```
magY = abs(Y)/max(abs(Y)); phaY = angle(Y)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0310e');
subplot(2,2,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,2); plot(w/pi,phaX,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.5:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase of X'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,3); plot(w/pi,magY,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = sort([-1:0.5:1]); magtick = [0:0.2:1.2];
xlabel('\omega/\pi','FontSize',12);
ylabel('|X|','FontSize',12);
title(['Scaled Magnitude from the
Property'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,2,4); plot(w/pi,phaY,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = sort([-1:0.4:1]); magtick = [-180:60:180];
xlabel('\omega/\pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase from the Property'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0310e;
```

The property verification using plots of $X\left(e^{jw}\right)$ is shown in Figure 3.23.

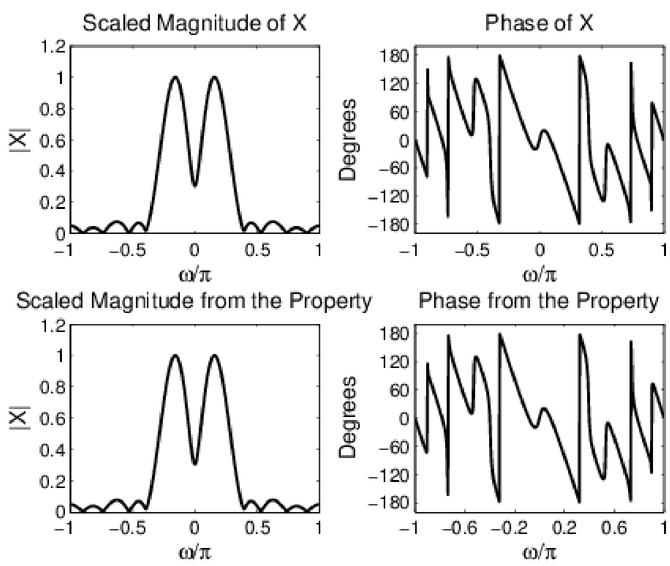


Figure 3.23: Problem P3.10e plots

P3.11

For each of the linear, shift-invariant systems described by the impulse response, determine the frequency response function $H(e^{jw})$. Plot the magnitude response $|H((e^{jw})|$ and the phase response $H((e^{jw})$ over the interval $[-\pi, \pi]$.

1. $h(n) = (0.9)^{|\mathbf{n}|}$
2. $h(n) = \text{sinc}(0.2n)[u(n + 20) - u(n - 20)]$, where sinc0 = 1.
3. $h(n) = \text{sinc}(0.2n)[u(n) - u(n - 40)]$
4. $h(n) = [(0.5)^n + (0.4)^n]u(n)$
5. $h(n) = (0.5)^{|\mathbf{n}|}\cos(0.1\pi n)$

Solutions

Determination and plots of the frequency response function $H(e^{jw})$

1. $h(n) = (0.9)^{|\mathbf{n}|}$

$$H\left(e^{j\omega}\right) = \sum_{h=-\infty}^{\infty}(n)e^{-jn\omega} = \sum_{0=-\infty}^{-1}.9^{-n}e^{-jn\omega} + \sum_{0=0}^{\infty}.9^{n}e^{-jn\omega} = \sum_{0=0}^{\infty}.9^{n}e^{jn\omega} - 1 + \sum_{0=0}^{\infty}.9^{n}e^{-jn\omega}$$

$$= \frac{1}{1-0.9e^{j\omega}} + \frac{1}{1-0.9e^{-j\omega}} - 1$$

$$= \frac{0.19}{1.81 - 1.8\cos(\omega)}$$

Matlab script:

```
% P3.11
%% P0311a: h(n) = (0.9)^|n|; H(w) = 0.19/(1.81-
1.8*cos(w));
clc; close all;
```

```
w = [-300:300]*pi/300;
H = 0.19*ones(size(w))./(1.81-1.8*cos(w));
magH = abs(H); phaH = angle(H)*180/pi;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0311a');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 20]);
wtick = [-1:0.2:1]; magtick = [0:5:20];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title('Magnitude response of h(n) =
(0.9)^{|n|}','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -180 180]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase response of h(n) =
(0.9)^{|n|}','FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0311a;
```

The magnitude and phase response plots of $H(e^{jw})$ are shown in Figure 3.24.

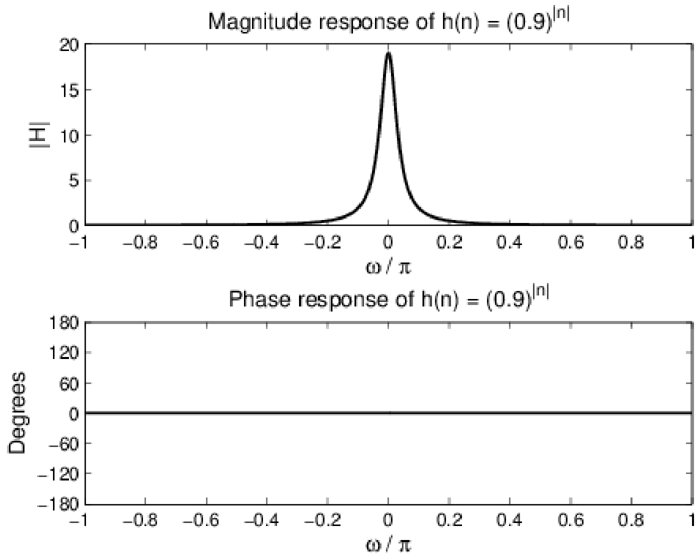


Figure 3.24: Problem P3.11a plots

2. $h(n) = \text{sinc}(0.2n)\,[u(n+20) - u(n-20)]$, where sinc 0 = 1.
Matlab script:

```
%% P0311b: h(n) = sinc(0.2*n)*[u(n+20)-u(n-20)]
```

```
clc; close all;
[h1,n1] = stepseq(-20,-20,20); [h2,n2] = stepseq(20,-
20,20);
[h3,n3] = sigadd(h1,n1,-h2,n2); n = n3; h =
sinc(0.2*n).*h3;
w = [-300:300]*pi/300; H = dtft(h,n,w); magH = abs(H);
phaH = angle(H)*180/pi;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0311b');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 6]);
wtick = [-1:0.2:1]; magtick = [0:1:6];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title(['Magnitude response of h(n) = sinc(0.2 \times
n)\times' ...
'[u(n+20)-u(n-20)]'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase response of h(n) = sinc(0.2 n)' ...
'\times[u(n+20)-u(n-20)]'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);set(gcf,'paperposi
tionmode','auto');
print -deps2 ../EPSFILES/P0311b;
```

The magnitude and phase response plots of $H(e^{jw})$ are shown in Figure 3.25.

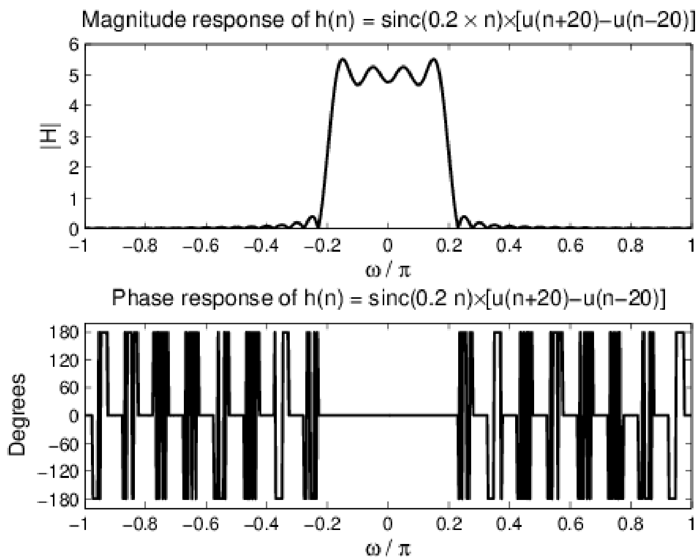


Figure 3.25: Problem P3.11b plots

3. $h(n) = \text{sinc}(0.2n)\,[u(n) - u\,(n - 40)]$

Matlab script:

```
%% P0311c: h(n) = sinc(0.2*n)*[u(n)-u(n-40)]
clc; close all;
[h1,n1] = stepseq(0,0,40); [h2,n2] = stepseq(40,0,40);
[h3,n3] = sigadd(h1,n1,-h2,n2); n = n3; h =
sinc(0.2*n).*h3;
w = [-300:300]*pi/300; H = dtft(h,n,w); magH = abs(H);
phaH = angle(H)*180/pi;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0311c');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 5]);
wtick = [-1:0.2:1]; magtick = [0:1:5];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title(['Magnitude response of h(n) =
sinc(0.2n)\times' ...
'[u(n)-u(n-40)]'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -180 180]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase response of h(n) = sinc(0.2)\times' ...
'[u(n)-u(n-40)]'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
set(gcf,'paperpositionmode','auto');print -
deps2 ../EPSFILES/P0311c;
```

The magnitude and phase response plots of $H(e^{jw})$ are shown in Figure 3.26.

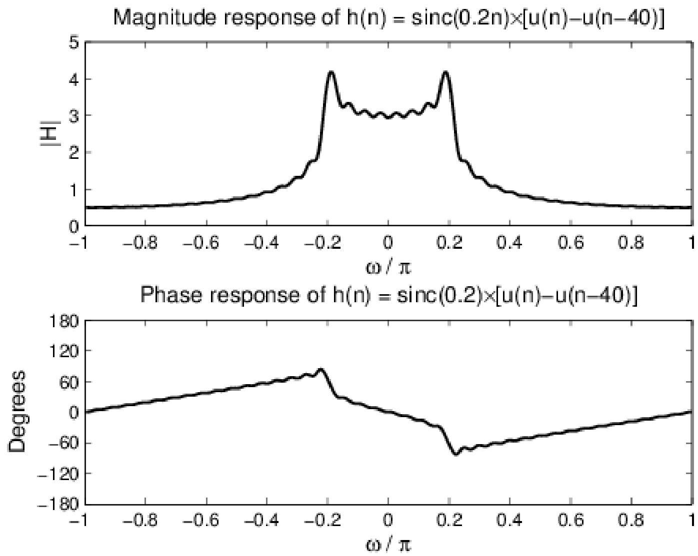


Figure 3.26: Problem P3.11c plots

4. $h(n) = \ (0.5)^n + \ (0.4)^n u(n)$

$$H\left(e^{j\omega}\right) = \sum_{h=-\infty}^{\infty}(n)e^{-jn\omega} = \sum_{0=0}^{\infty}.5^n e^{-jn\omega} + \sum_{0=0}^{\infty}.4^n e^{-jn\omega} = \frac{1}{1-0.5e^{-j\omega}} + \frac{1}{1-0.4e^{-j\omega}}$$

$$= \frac{2-0.9e^{-j\omega}}{1-0.9e^{-j\omega}+0.2e^{-j2\omega}}$$

Matlab script:

```
%% P0311d: h(n) = ((0.5)^n+(0.4)^n) u(n);
% H(w) = (2-0.9*exp(-j*w))./(1-0.9*exp(-j*w)+0.2*exp(-
j*2*w))
clc; close all;
w = [-300:300]*pi/300; H = (2-0.9*exp(-1i*w))./(1-
0.9*exp(-1i*w)+0.2*exp(-1i*2*w));
magH = abs(H); phaH = angle(H)*180/pi;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0311d');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 1 4]);
wtick = [-1:0.2:1]; magtick = [0:0.5:4];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title('Magnitude response:h(n) = [(0.5)^n+(0.4)^n]
u(n)','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -180 180]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
```

```
title('Phase response:h(n) = [(0.5)^n+(0.4)^n]
u(n)','FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);set(gcf,'paperposi
tionmode','auto');
print -deps2 ../EPSFILES/P0311d;
```

The magnitude and phase response plots of $H(e^{jw})$ are shown in Figure 3.27.

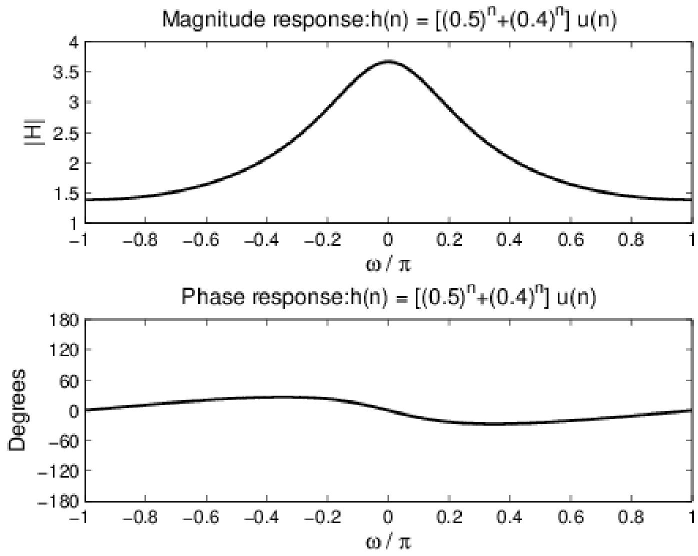


Figure 3.27: Problem P3.11d plots

5. $h(n) = (0.5)^{|n|}\cos(0.1\pi n) = \frac{1}{2}0.5^{|n|}e^{j0.1\pi n} + \frac{1}{2}0.5^{|n|}e^{-j0.1\pi n}$

$$H\left(e^{j\omega}\right) = \sum_{h=-\infty}^{\infty} (n)e^{-jn\omega} = \frac{1}{2}\left[\sum_{0=-\infty}^{\infty} .5^{|n|}e^{-j(\omega-0.1\pi)n} + \sum_{0=-\infty}^{\infty} .5^{|n|}e^{-j(\omega+0.1\pi)n}\right]$$

$$= \frac{0.5\times 0.75}{1.25-\cos(\omega-0.1\pi)} + \frac{0.5\times 0.75}{1.25-\cos(\omega+0.1\pi)}$$

Matlab script:

```
%% P0311e: h(n) = (0.5)^|n|*cos(0.1*pi*n);
% H(w) = 0.5*0.75*ones(size(w)) ./(1.25-cos(w-(0.1*pi)))+
% 0.5*0.75*ones(size(w)) ./(1.25-cos(w+(0.1*pi)))
clc; close all;
w = [-300:300]*pi/300; H = 0.5*0.75*ones(size(w))./(1.25-
cos(w-(0.1*pi)))+...
0.5*0.75*ones(size(w))./(1.25-cos(w+(0.1*pi)));
magH = abs(H); phaH = angle(H)*180/pi;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0311e');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 3]);
wtick = [-1:0.2:1]; magtick = [0:0.5:3];
```

```
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title(['Magnitude response'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -180 180]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase response'],'FontSize',12);
set(gca,'XTick',wtick,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0311e;
```

The magnitude and phase response plots of $H(e^{jw})$ are shown in Figure 3.28.

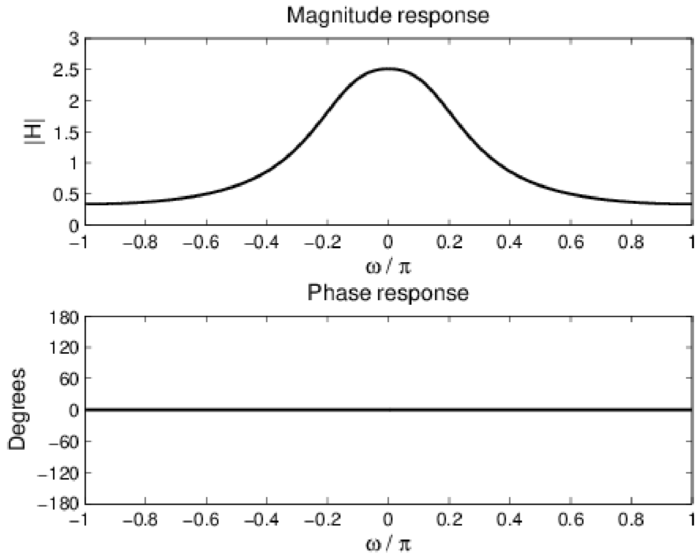


Figure 3.28: Problem P3.11e plots

P3.12

Let $x(n) = A\cos(\omega_0 n + \theta_0)$ be an input sequence to an LTI system described by the impulse response $h(n)$. Show that the output sequence $y(n)$ is given by

$$y(n) = A|H(e^{j\omega_0})|\cos[\omega_0 n + \theta_0 + \angle H(e^{j\omega_0})]$$

Solutions

$$
\begin{aligned}
y(n) = h(n) * x(n) &= A \sum_{h=-\infty}^{\infty} (k)\cos[\omega_0(n-k)+\theta_0] \\
&= \frac{A}{2} \sum_{h=-\infty}^{\infty} (k)\exp\left[j\omega_0(n-k)+\theta_0\right] + \frac{A}{2} \sum_{h=-\infty}^{\infty} (k)\exp\left[-j\omega_0(n-k)-\theta_0\right] \\
&= \frac{A}{2} e^{j\theta_0}\left[\sum_{h=-\infty}^{\infty} (k)e^{-j\omega_0 k}\right] e^{j\omega_0 n} + \frac{A}{2} e^{-j\theta_0}\left[\sum_{h=-\infty}^{\infty} (k)e^{j\omega_0 k}\right] e^{-j\omega_0 n} \\
&= \frac{A}{2} e^{j\theta_0} H\left(e^{j\omega_0}\right) e^{j\omega_0 n} + \frac{A}{2} e^{-j\theta_0} H^*\left(e^{j\omega_0}\right) e^{-j\omega_0 n} \\
&= \frac{A}{2} e^{j\theta_0} |H\left(e^{j\omega_0}\right)| e^{j\angle H\left(e^{j\omega_0}\right)} e^{j\omega_0 n} + \frac{A}{2} e^{-j\theta_0} |H\left(e^{j\omega_0}\right)| e^{-j\angle H\left(e^{j\omega_0}\right)} e^{-j\omega_0 n} \\
&= \frac{A}{2} |H\left(e^{j\omega_0}\right)| \left[\exp\left\{j\left[\omega_0 n + j\theta_0 + j\angle H\left(e^{j\omega_0}\right)\right]\right\} + \exp\left\{-j\left[\omega_0 n + j\theta_0 + j\angle H\left(e^{j\omega_0}\right)\right]\right\}\right] \\
&= A|H\left(e^{j\omega_0}\right)| \cos\left[\omega_0 n + j\theta_0 + j\angle H\left(e^{j\omega_0}\right)\right]
\end{aligned}
$$

P3.13

Let $x(n) = 3\cos(0.5\pi n + 60°) + 2\sin(0.3\pi n)$ be the input to each of the systems described in Problem P3.11. In each case, determine the output sequence $y(n)$.

Solutions

Sinusoidal steady-state responses

1. The input to the system $h(n) = (0.9)^{|n|}$ is $x(n) = 3\cos(0.5\pi n + 60°) + 2\sin(0.3\pi n)$. The steady-state response $y(n)$ is computed using Matlab.

```
% P3.13
%% P0313a: x(n) = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
i.e.
% x(n) = 3*cos(0.5*pi*n+pi/3)+2*cos(0.3*pi*n-pi/2)
% h(n) = (0.9)^|n|
clc; close all;
%
n = [-20:20]; x = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
w1 = 0.5*pi; H1 = 0.19*w1/(1.81-1.8*cos(w1));
w2 = 0.3*pi; H2 = 0.19*w2/(1.81-1.8*cos(w2));
magH1 = abs(H1); phaH1 = angle(H1); magH2 = abs(H2);
phaH2 = angle(H2);
y = 3*magH1*cos(w1*n+pi/3+phaH1)+...
2*magH2*cos(w2*n-pi/2+phaH2);
```

```
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0313a');
subplot(2,1,1); Hs = stem(n,x,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
axis([-22 22 -5 5]);
title(['Input sequence x(n): 3*cos(0.5{\pi}n + \pi/3) +
2sin(0.3{\pi}n)'],...
'FontSize',12);
subplot(2,1,2); Hs = stem(n,y,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
axis([-22 22 -5 5]);
title('Output sequence y(n) for h(n) =
(0.9)^{|n|}','FontSize',12);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0313a;
```

The magnitude and phase response plots of $H(e^{jw})$ are shown in Figure 3.29.

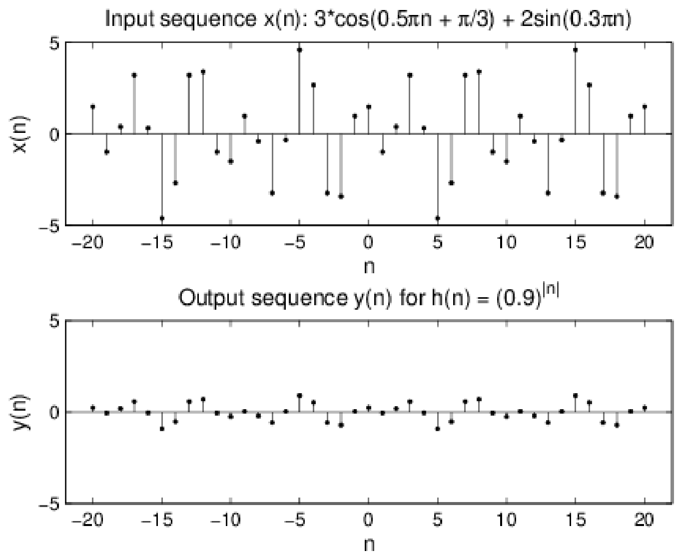


Figure 3.29: Problem P3.13a plots

2. The input to the system $h(n) = \mathrm{sinc}(0.2n)\,[u\,(n + 20) - u\,(n - 20)]$, where sinc 0 = 1. The steadystate response $y(n)$ is computed using Matlab.

```
%% P0313b: x(n) = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
i.e.
% x(n) = 3*cos(0.5*pi*n+pi/3)+2*cos(0.3*pi*n-pi/2)
% h(n) = sinc(0.2*n)*[u(n+20)-u(n-20)]
clc; close all;
%
```

```
n = [-20:20]; x = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
[h1,n1] = stepseq(-20,-20,20); [h2,n2] = stepseq(20,-
20,20);
[h3,n3] = sigadd(h1,n1,-h2,n2); n = n3; h =
sinc(0.2*n).*h3;
w1 = 0.5*pi; H1 = dtft(h,n,w1); w2 = 0.3*pi; H2 =
dtft(h,n,w2);
magH1 = abs(H1); phaH1 = angle(H1); magH2 = abs(H2);
phaH2 = angle(H2);
y = 3*magH1*cos(w1*n+pi/3+phaH1)+2*magH2*cos(w2*n-
pi/2+phaH2);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0313b');
subplot(2,1,1); Hs = stem(n,x,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
axis([-22 22 -5 5]);
title(['Input sequence x(n): 3*cos(0.5{\pi}n + \pi/3) +
2sin(0.3{\pi}n)'],...
'FontSize',12);
subplot(2,1,2); Hs = stem(n,y,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
axis([-22 22 -5 5]);
title('Output sequence y(n) for h(n) = sinc(0.2
n)[u(n+20) - u(n-20)]',...
'FontSize',12); set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0313b;
```

The magnitude and phase response plots of $H\left(e^{jw}\right)$ are shown in Figure 3.30.

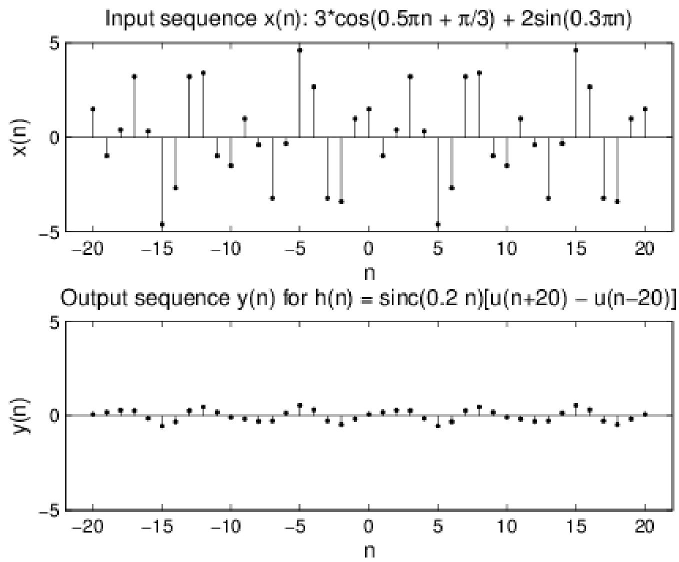


Figure 3.30: Problem P3.13b plots

3. The input to the system $h(n)$ = sinc(0.2n) [$u(n)$ - u (n - 40)]. The steady-state response $y(n)$ is computed using Matlab.

```
%% P0313c: x(n) = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
i.e.
% x(n) = 3*cos(0.5*pi*n+pi/3)+2*cos(0.3*pi*n-pi/2)
% h(n) = sinc(0.2*n)*[u(n)-u(n-40)]
clc; close all;
%
n = [-20:20]; x = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
[h1,n1] = stepseq(0,0,40); [h2,n2] = stepseq(40,0,40);
[h3,n3] = sigadd(h1,n1,-h2,n2); h = sinc(0.2*n3).*h3;
w1 = 0.5*pi; w2 = 0.3*pi; H1 = dtft(h,n3,w1); H2 =
dtft(h,n3,w2);
magH1 = abs(H1); phaH1 = angle(H1); magH2 = abs(H2);
phaH2 = angle(H2);
y = 3*magH1*cos(w1*n+pi/3+phaH1)+2*magH2*cos(w2*n-
pi/2+phaH2);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0313c');
subplot(2,1,1); Hs = stem(n,x,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
axis([-22 22 -5 5]);
title(['Input sequence x(n): 3*cos(0.5{\pi}n + \pi/3) +
2sin(0.3{\pi}n)'],...
'FontSize',12);
subplot(2,1,2); Hs = stem(n,y,'filled');
```

```
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
axis([-22 22 -5 5]);
title('Output sequence y(n) for h(n) = sinc(0.2 n)[u(n) -
u(n-40)]',...
'FontSize',12);set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0313c;
```

The magnitude and phase response plots of $H\left(e^{jw}\right)$ are shown in Figure 3.31.

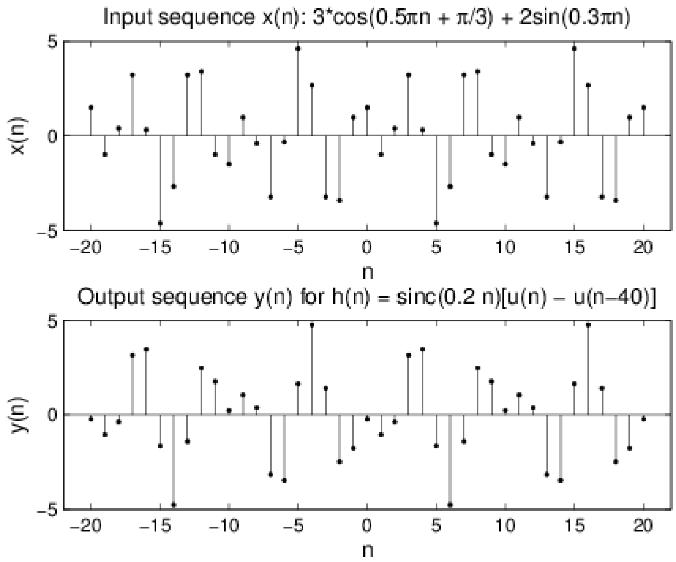


Figure 3.31: Problem P3.13c plots

4. The input to the system $h(n) = (0.5)^{\boldsymbol{n}} + (0.4)^{\boldsymbol{n}}\ u(n)$. The steady-state response $y(n)$ is computed using Matlab.

```
%% P0313d: x(n) = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
i.e.
% x(n) = 3*cos(0.5*pi*n+pi/3)+2*cos(0.3*pi*n-pi/2)
% h(n) = ((0.5)^(n)+(0.4)^(n)).*u(n)
clc; close all;
%
n = [-20:20]; x = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
w1 = 0.5*pi; H1 = (2-0.9*exp(-1i*w1))./(1-0.9*exp(-
1i*w1)+0.2*exp(-1i*2*w1));
w2 = 0.3*pi; H2 = (2-0.9*exp(-1i*w2))./(1-0.9*exp(-
1i*w2)+0.2*exp(-1i*2*w2));
magH1 = abs(H1); phaH1 = angle(H1); magH2 = abs(H2);
phaH2 = angle(H2);
y = 3*magH1*cos(w1*n+pi/3+phaH1)+2*magH2*cos(w2*n-
pi/2+phaH2);
%
```

```
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0313d');
subplot(2,1,1); Hs = stem(n,x,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
axis([-22 22 -10 10]);
title(['Input sequence x(n): 3*cos(0.5{\pi}n + \pi/3) +
2sin(0.3{\pi}n)'],...
'FontSize',12);
subplot(2,1,2); Hs = stem(n,y,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
axis([-22 22 -10 10]);
title('Output sequence y(n) for h(n) = [(0.5)^n+(0.4)^n]
u(n)]',...
'FontSize',12); set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0313d;
```

The magnitude and phase response plots of $H\left(e^{jw}\right)$ are shown in Figure 3.32.

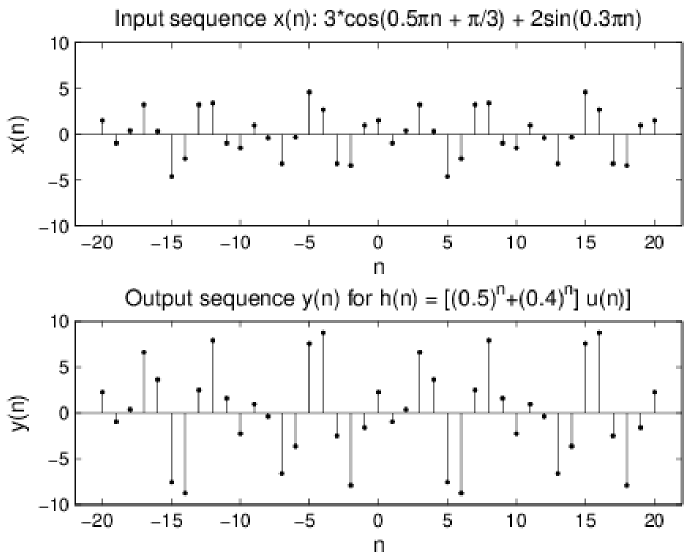


Figure 3.32: Problem P3.13d plots

5. The input to the system $h(n) = (0.5)^{|n|}\cos(0.1\pi n)$. The steady-state response $y(n)$ is computed using Matlab.

```
%% P0313e: x(n) = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
i.e.
% x(n) = 3*cos(0.5*pi*n+pi/3)+2*cos(0.3*pi*n-pi/2)
% h(n) = (0.5)^|n|*cos(0.1*pi*n);
clc; close all;
%
n = [-20:20]; x = 3*cos(0.5*pi*n+pi/3)+2*sin(0.3*pi*n);
```

```
w1 = 0.5*pi; H1 = 0.5*0.75*w1/(1.25-cos(w1-(0.1*pi)))+...
0.5*0.75*w1/(1.25-cos(w1+(0.1*pi)));
w2 = 0.3*pi; H2 = 0.5*0.75*w2/(1.25-cos(w2-(0.1*pi)))+...
0.5*0.75*w2/(1.25-cos(w2+(0.1*pi)));
magH1 = abs(H1); phaH1 = angle(H1); magH2 = abs(H2);
phaH2 = angle(H2);
y = 3*magH1*cos(w1*n+pi/3+phaH1)+2*magH2*cos(w2*n-
pi/2+phaH2);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0313e');
subplot(2,1,1); Hs = stem(n,x,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
axis([-22 22 -6 6]);
title(['Input sequence x(n): 3*cos(0.5{\pi}n + \pi/3) +
2sin(0.3{\pi}n)'],...
'FontSize',12);
subplot(2,1,2); Hs = stem(n,y,'filled');
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
axis([-22 22 -6 6]);
title('Output sequence y(n) for h(n) =
(0.5)^{|n|}cos(0.1{\pi}n)] u(n)]',...
'FontSize',12); set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0313e;
```

The magnitude and phase response plots of $H(e^{jw})$ are shown in Figure 3.33.

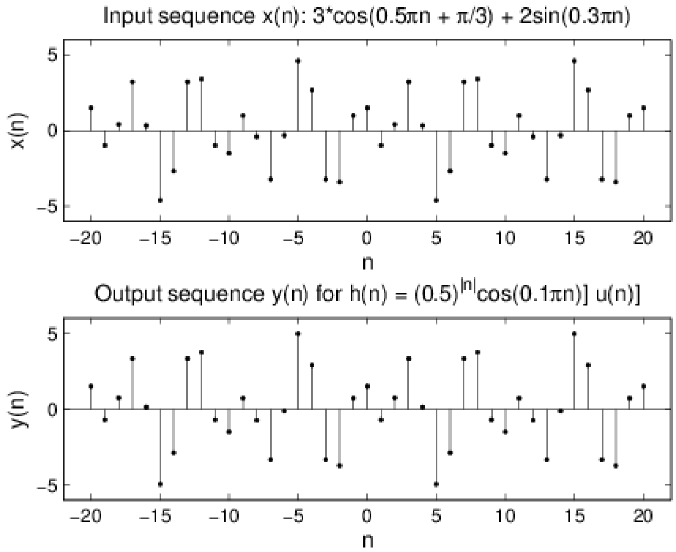


Figure 3.33: Problem P3.13e plots

P3.14

An ideal lowpass filter is described in the frequency domain by

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\alpha\omega}, & |\omega| \le \omega_c \\ 0, & \omega_c < |\omega| \le \pi \end{cases}$$

where ω_c is called the cutoff frequency and α is called the phase delay.
1. Determine the ideal impulse response $h_d(n)$ using the IDTFT relation (3.2).
2. Determine and plot the truncated impulse response

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

for $N = 41$, $\alpha = 20$, and $\omega_c = 0.5\pi$.
3. Determine and plot the frequency response function $H(e^{jw})$, and compare it with the ideal lowpass filter response $H_d(e^{jw})$. Comment on your observations.

Solutions

1. The ideal impulse response $h_d(n)$ using the IDTFT relation (3.2):

$$h_d(n) = \mathcal{F}^{-1}\left[H_d(e^{j\omega})\right] = \frac{1}{2\pi}\int_{-\infty}^{\infty} H_d(e^{j\omega})e^{jn\omega}d\omega = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{-j\alpha\omega}e^{jn\omega}d\omega = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j(n-\alpha)\omega}d\omega$$

$$= \frac{1}{2\pi}\left.\frac{e^{j(n-\alpha)\omega}}{j(n-\alpha)}\right|_{-\omega_c}^{\omega_c} = \frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}$$

2. Plot of the truncated impulse response:

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} = \begin{cases} \dfrac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

for $N = 41$, $\alpha = 20$, and $\omega_c = 0.5\pi$. Matlab script:

```
% P3.14
%% P0314b: Truncated Ideal Lowpass Filter; h(n) =
h_d(n) ,0 <= n <= N-1
% = 0 ,otherwise
clc; close all;
%
n = [0:40]; alpha = 20; wc = 0.5*pi;
fc = wc/(2*pi); h = 2*fc*sinc(2*fc*(n-alpha));
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0314b');
Hs = stem(n,h,'filled'); set(Hs,'markersize',2); axis([-2
42 -0.2 0.6]);
xlabel('n','FontSize',12); ylabel('h(n)','FontSize',12);
title('Truncated Impulse Response h(n)','FontSize',12);
```

```
set(gca,'YTick',[-0.2:0.1:0.6]);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0314b;
```

The truncated impulse response plot of $h_d(n)$ is shown in Figure 3.34.

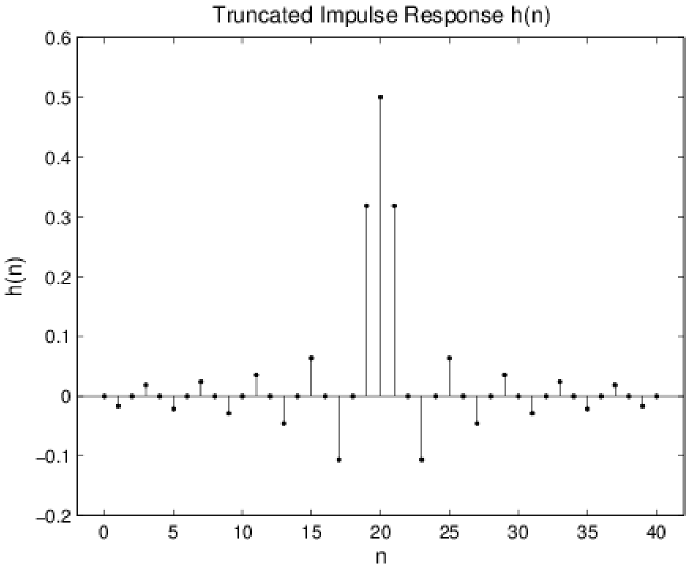


Figure 3.34: Problem P3.14b plot

3. Plot of the frequency response function $H\left(e^{jw}\right)$ and comparison with the ideal lowpass filter response $H_d\left(e^{jw}\right)$: Matlab script:

```
%% P0314c: Freq Resp of truncated and ideal impulse
responses for lowpass filter
clc; close all;
%
K = 500; w = [-K:K]*pi/K; H = dtft(h,n,w); magH = abs(H);
phaH = angle(H);
H_d = zeros(1,length(w)); H_d(K/2+1:3*K/2+1) = exp(-
1i*alpha*w(K/2+1:3*K/2+1));
magH_d = abs(H_d); phaH_d = angle(H_d); wtick = sort([-
1:0.4:1 0]);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0314c');
subplot(2,2,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 1.2]);
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title('Magnitude of H(e^{j\omega})','FontSize',12);
set(gca,'XTick',wtick);
subplot(2,2,2); plot(w/pi,phaH*180/pi,'LineWidth',1.5);
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase of H(e^{j\omega})','FontSize',12);
```

```
set(gca,'XTick',wtick); magtick = [-180:60:180];
set(gca,'YTick',magtick); set(gca,'XTick',wtick);
subplot(2,2,3); plot(w/pi,magH_d,'LineWidth',1.5);
axis([-1 1 0 1.2]);
xlabel('\omega / \pi','FontSize',12);
ylabel('|H_d|','FontSize',12);
title('Magnitude of H_d(e^{j\omega})','FontSize',12);
set(gca,'XTick',wtick); ytick = [0:0.2:1.2];
set(gca,'YTick',ytick);
subplot(2,2,4); plot(w/pi,phaH_d*180/pi,'LineWidth',1.5);
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase of H_d(e^{j\omega})','FontSize',12);
set(gca,'XTick',wtick); magtick = [-180:60:180];
set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0314c;
```

The frequency responses are shown in Figure 3.35 from which we observe that the truncated response is a smeared or blurred version of the ideal response.

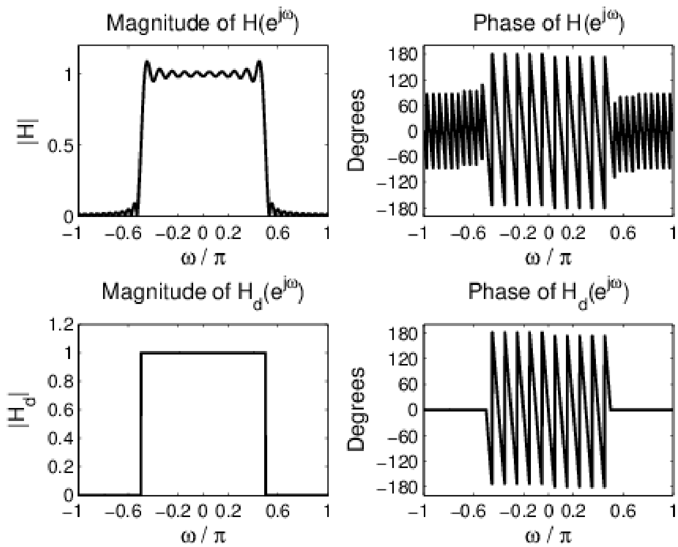


Figure 3.35: Problem P3.14c plots

P3.15

An ideal highpass filter is described in the frequency-domain by

$$H_d(e^{j\omega}) = \begin{cases} 1 \cdot e^{-j\alpha\omega}, & \omega_c < |\omega| \le \pi \\ 0, & |\omega| \le \omega_c \end{cases}$$

where ω_c is called the cutoff frequency and α is called the phase delay.

1. Determine the ideal impulse response $h_d(n)$ using the IDTFT relation (3.2)
2. Determine and plot the truncated impulse response

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

for $N = 31$, $\alpha = 15$, and $\omega_c = 0.5\pi$.
3. Determine and plot the frequency response function $H(e^{jw})$, and compare it with the ideal highpass filter response $H_d(e^{jw})$. Comment on your observations

Solutions

1. The ideal impulse response $h_d(n)$ using the IDTFT relation (3.2):

$$h_d(n) = \mathcal{F}^{-1}\left[H_d(e^{j\omega})\right] = \frac{1}{2\pi}\int_{-\infty}^{\infty} H_d(e^{j\omega})e^{jn\omega}d\omega = \frac{1}{2\pi}\int_{-\pi}^{-\omega_c} e^{-j\alpha\omega}e^{jn\omega}d\omega + \frac{1}{2\pi}\int_{\omega_c}^{\pi} e^{-j\alpha\omega}e^{jn\omega}d\omega$$
$$= \frac{1}{2\pi}\int_{-\pi}^{\pi} e^{j(n-\alpha)\omega}d\omega - \frac{1}{2\pi}\int_{-\omega_c}^{\omega} e^{j(n-\alpha)\omega}d\omega = \frac{\sin[\pi(n-\alpha)]}{\pi(n-\alpha)} - \frac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}$$

2. Plot of the truncated impulse response:

$$h(n) = \begin{cases} h_d(n), & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases} = \begin{cases} \dfrac{\sin[\pi(n-\alpha)]}{\pi(n-\alpha)} - \dfrac{\sin[\omega_c(n-\alpha)]}{\pi(n-\alpha)}, & 0 \le n \le N-1 \\ 0, & \text{otherwise} \end{cases}$$

for $N = 31$, $\alpha = 15$, and $\omega_c = 0.5\pi$. Matlab script:

```
% P3.15
%% P0315b: Ideal Highpass Filter; h(n) = h_d(n) ,0 <= n
<= N-1
% = 0 ,otherwise
clc; close all;
n = [0:40]; alpha = 20; wc = 0.5*pi; fc = wc/(2*pi);
h = sinc(n-alpha)-2*fc*sinc(2*fc*(n-alpha));
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0315b');
Hs = stem(n,h,'filled'); set(Hs,'markersize',2); axis([-2
42 -0.4 0.6]);
xlabel('n','FontSize',12); ylabel('h(n)','FontSize',12);
title('Truncated Impulse Response h(n)','FontSize',12);
set(gca,'YTick',[-0.4:0.1:0.6]);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0315b;
```

The truncated impulse response plot of $h_d(n)$ is shown in Figure 3.36.

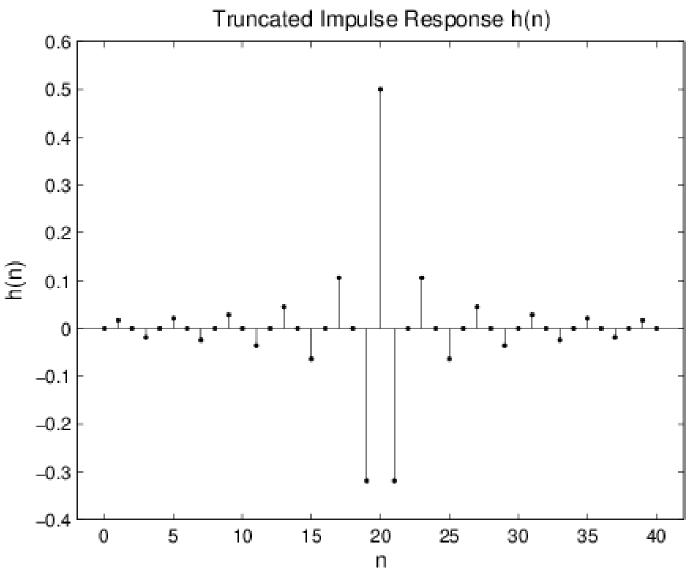


Figure 3.36: Problem P3.15b plot

3. Plot of the frequency response function $H(e^{jw})$and comparison with the ideal lowpass filter response $H_d\left(e^{jw}\right)$: Matlab script:

```
%% P0315c: Freq Resp of truncated and ideal impulse
responses for highpass filter
clc; close all;
K = 500; w = [-K:K]*pi/K; H = dtft(h,n,w); magH = abs(H);
phaH = angle(H);
H_d = zeros(1,length(w)); H_d(1:K/2+1) = exp(-
1i*alpha*w(1:K/2+1));
H_d(3*K/2+1:end) = exp(-1i*alpha*w(3*K/2+1:end));
magH_d = abs(H_d); phaH_d = angle(H_d); wtick = sort([-
1:0.4:1 0]);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0315c');
subplot(2,2,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 1.2]);
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title('Magnitude of H(e^{j\omega})','FontSize',12);
set(gca,'XTick',wtick); set(gca,'XTick',wtick);
subplot(2,2,2); plot(w/pi,phaH*180/pi,'LineWidth',1.5);
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase of H(e^{j\omega})','FontSize',12);
set(gca,'XTick',wtick); magtick = [-180:60:180];
set(gca,'YTick',magtick);
subplot(2,2,3); plot(w/pi,magH_d,'LineWidth',1.5);
axis([ -1 1 0 1.2]);
xlabel('\omega / \pi','FontSize',12);
```

```
ylabel('|H_d|','FontSize',12);
title('Magnitude of H_d(e^{j\omega})','FontSize',12);
set(gca,'XTick',wtick); ytick =
[0:0.2:1.2];set(gca,'YTick',ytick);
subplot(2,2,4); plot(w/pi,phaH_d*180/pi,'LineWidth',1.5);
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase of H_d(e^{j\omega})','FontSize',12);
set(gca,'XTick',wtick); magtick = [-180:60:180];
set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0315c;
```

The frequency responses are shown in Figure 3.37 from which we observe that the truncated response is a smeared or blurred version of the ideal response.

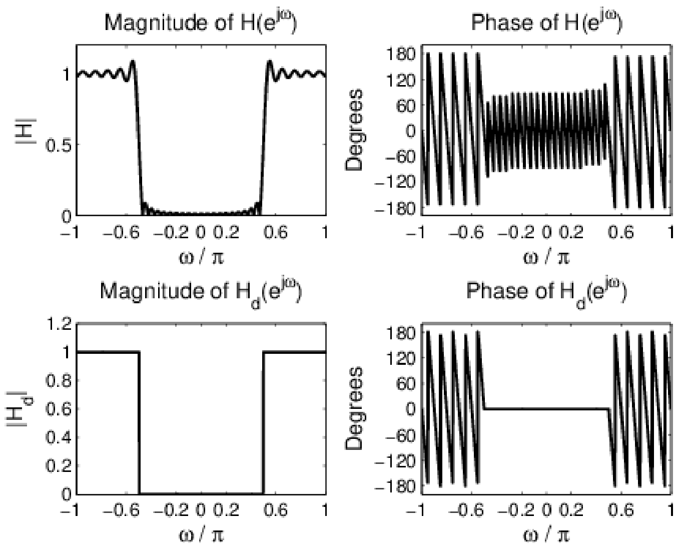


Figure 3.37: Problem P3.15c plots

P3.16

For a linear, shift-invariant system described by the difference equation

$$y(n) = \sum_{m=0}^{M} b_m x(n-m) - \sum_{\ell=1}^{N} a_\ell y(n-\ell)$$

the frequency-response function is given by

$$H(e^{j\omega}) = \frac{\sum_{m=0}^{M} b_m e^{-j\omega m}}{1 + \sum_{\ell=1}^{N} a_\ell e^{-j\omega \ell}}$$

Write a MATLAB function **freqresp** to implement this relation. The format of this function should be

function [H] = freqresp(b,a,w)
% Frequency response function from difference equation
% [H] = freqresp(b,a,w)
% H = frequency response array evaluated at w frequencies
% b = numerator coefficient array
% a = denominator coefficient array (a(1)=1)
% w = frequency location array

Solutions

Matlab function **freqresp**.

```
function [H] = freqresp(b,a,w)
% Frequency response function from difference equation
% [H] = freqresp(b,a,w)
% H = frequency response array evaluated at w frequencies
% b = numerator coefficient array
% a = denominator coeeficient array (a(1) = 1)
% w = frequency location array
%
b = reshape(b,1,length(b));
a = reshape(a,1,length(a));
w = reshape(w,1,length(w));
m = 0:length(b)-1;num = b*exp(-1i*m'*w);
l = 0:length(a)-1;den = a*exp(-1i*l'*w);
H = num./den;
```

P3.17

Determine *H*(*ejω*), and plot its magnitude and phase for each of the following systems:

1. $y(n) = \frac{1}{5}\sum_{m=0}^{4} x(n-m)$
2. $y(n) = x(n) - x(n-2) + 0.95y(n-1) - 0.9025y(n-2)$
3. $y(n) = x(n) - x(n-1) + x(n-2) + 0.95y(n-1) - 0.9025y(n-2)$
4. $y(n) = x(n) - 1.7678x(n-1) + 1.5625x(n-2) + 1.1314y(n-1) - 0.64y(n-2)$
5. $y(n) = x(n) - \sum_{\ell=1}^{5} (0.5)^{\ell}\, y(n-\ell)$

Solutions

1. $y(n) = \frac{1}{5}\sum_{m=0}^{4} x(n-m)$:

Matlab script:

```
% P3.17
%% P0317a: y(n) = (1/5) sum_{0}^{4} x(n-m)
clc; close all;
%
w = [-300:300]*pi/300; a = [1]; b = [0.2 0.2 0.2 0.2
0.2];
[H] = freqresp(b,a,w); magH = abs(H); phaH =
angle(H)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0317a');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 1.2]);
wtick = [-1:0.2:1]; magtick = [0:0.2:1.2];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title(['Magnitude response'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -220 220]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase Response ','FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0317a;
```

The frequency responses are shown in Figure 3.38

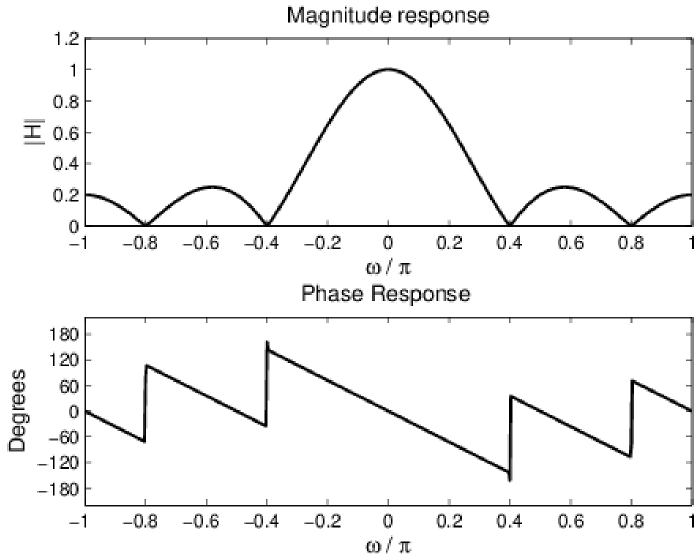


Figure 3.38: Frequency response plots in Problem P3.17a

2. $y(n) = x(n) - x(n - 2) + 0.95y(n - 1) - 0.9025y(n - 2)$

Matlab script:

```
%% P0317b: y(n) = x(n)-x(n-2)+0.95*y(n-1)-0.9025*y(n-2)
clc; close all;
%
w = [-300:300]*pi/300; a = [1 -0.95 0.9025]; b = [1 0 -
1];
[H] = freqresp(b,a,w); magH = abs(H); phaH =
angle(H)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0317b');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 25]);
wtick = [-1:0.2:1]; magtick = [0:5:25];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title(['Magnitude response'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase response'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0317b;
```

The frequency responses are shown in Figure 3.39

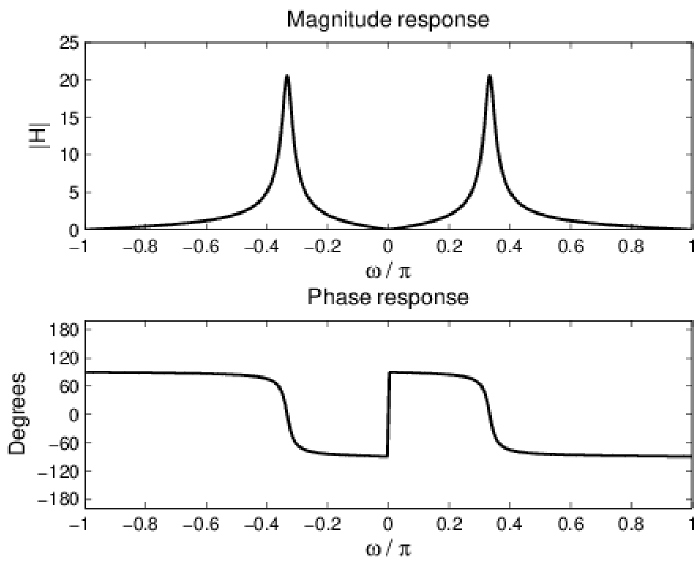


Figure 3.39: Frequency response plots in Problem P3.17b

3. $y(n) = x(n) - x(n - 1) + x(n - 2) + 0.95y(n - 1) - 0.9025y(n - 2)$

Matlab script:

```
%% P0317c: y(n)  = x(n)-x(n-1)+x(n-2)+0.95*y(n-1)-
0.9025*y(n-2)
clc; close all;
%
w = [-300:300]*pi/300; a = [1 -0.95 0.9025]; b = [1 -1
1];
[H] = freqresp(b,a,w); magH = abs(H); phaH =
angle(H)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0317c');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 1.4]);
wtick = [-1:0.2:1]; magtick = [0:0.2:1.4];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title(['Magnitude response'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase response'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
```

```
print -deps2 ../EPSFILES/P0317c;
```

The frequency responses are shown in Figure 3.40

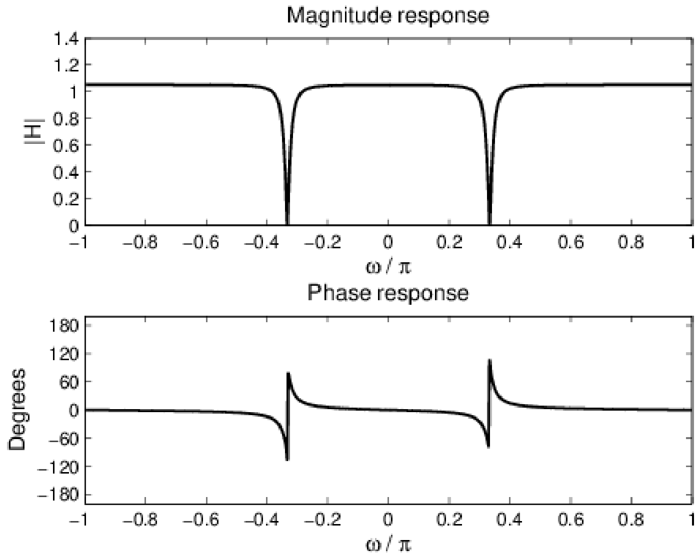


Figure 3.40: Frequency response plots in Problem P3.17c

4. $y(n) = x(n) - 1.7678x(n - 1) + 1.5625x(n - 2) + 1.1314y(n - 1) - 0.64y(n - 2)$
Matlab script:

```
%% P0317d: y(n) = x(n)-1.7678*x(n-1)+1.5625*x(n-
2)+1.1314*y(n-1)- 0.64*y(n-2)
clc; close all;
%
w = [-300:300]*pi/300; a = [1 -1.1314 0.64]; b = [1 -
1.7678 1.5625];
[H] = freqresp(b,a,w); magH = abs(H); phaH =
angle(H)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0317d');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5);
wtick = [-1:0.2:1]; magtick = [1.5:0.02:1.6];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title(['Magnitude response'],'FontSize',12);
set(gca,'XTick',wtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -200 200]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase response','FontSize',12);
```

```
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0317d;
```

The frequency responses are shown in Figure 3.41

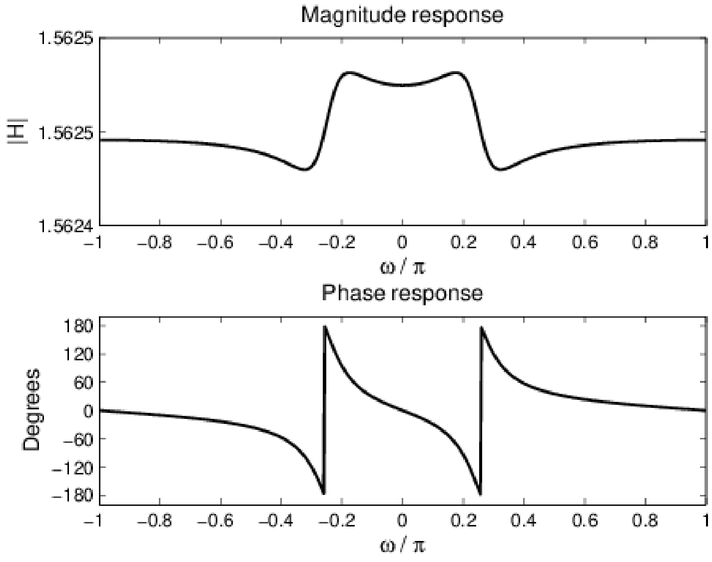


Figure 3.41: Frequency response plots in Problem P3.17d

5. $y(n) = x(n) - \sum_{l=1}^{5}(0.5)^{l}\, y\,(n\; - \;\ell)$
Matlab script:

```
%% P0317e: y(n) = x(n)-sum _ {l = 1} ^ {5} (0.5) ^ l*y(n-
l);
clc; close all;
%
w = [-300:300]*pi/300; l = [0:5]; a = 0.5 .^ l; b = [1];
[H] = freqresp(b,a,w);
magH = abs(H); phaH = angle(H)*180/pi;
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0317e');
subplot(2,1,1); plot(w/pi,magH,'LineWidth',1.5); axis([-1
1 0 1.8]);
wtick = [-1:0.2:1]; magtick = [0:0.2:1.8];
xlabel('\omega / \pi','FontSize',12);
ylabel('|H|','FontSize',12);
title(['Magnitude Response'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaH,'LineWidth',1.5); axis([-1
1 -180 180]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
```

```
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title(['Phase Response'],'FontSize',12);
set(gca,'XTick',wtick); set(gca,'YTick',magtick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0317e;
```

The frequency responses are shown in Figure 3.42

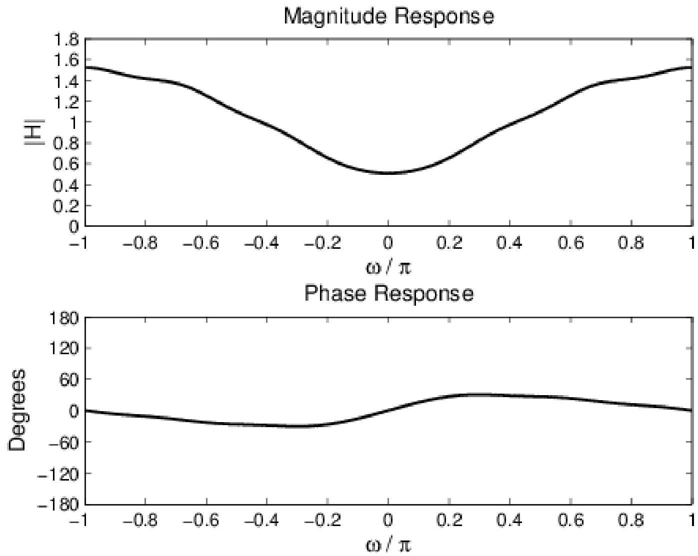


Figure 3.42: Frequency response plots in Problem P3.17e

P3.18

A linear, shift-invariant system is described by the difference equation

$$y(n) = \sum_{m=0}^{3} x(n-2m) - \sum_{\ell=1}^{3} (0.81)^{\ell}\, y(n-2\ell)$$

Determine the steady-state response of the system to the following inputs:

1. $x(n) = 5 + 10\ \ (-1)^n$
2. $x(n) = 1 + \cos(0.5\pi n + \pi/2)$
3. $x(n) = 2\sin(\pi n/4) + 3\cos(3\pi n/4)$
4. $x(n) = \ \sum_{k=0}^{5}(k\ +\ 1)\cos(\pi kn/4)$
5. $x(n) = \cos(\pi n)$

In each case, generate $x(n)$, $0 \le n \le 200$, and process it through the **filter** function to obtain $y(n)$. Compare your $y(n)$ with the steady-state responses in each case.

Solutions

A linear, shift-invariant system is described by the difference equation

$$y(n)=\sum_{m=0}^{3}x(n-2m)-\sum_{\ell=1}^{3}(0.81)^{\ell}\,y(n-2\ell)\Rightarrow H\left(e^{j\omega}\right)=\frac{\sum_{m=0}^{3}e^{-j2m\omega}}{\sum_{\ell=0}^{3}(0.81)^{2\ell}e^{-j2\ell\omega}}$$

1. $x(n) = 5 + 10(-1)^n = 5 + 10\cos(n\pi)$: We need frequency responses at ω = 0 and ω = π.

$$H\left(e^{j0}\right)=\frac{\sum_{m=0}^{3}e^{-j0}}{\sum_{\ell=0}^{3}(0.81)^{2\ell}e^{-j0}}=1.6885 \text{ and } H\left(e^{j\pi}\right)=\frac{\sum_{m=0}^{3}e^{-j2\pi\ell}}{\sum_{\ell=0}^{3}(0.81)^{2\ell}e^{-j2\pi\ell}}=1.6885$$

Hence the steady-state response is $y(n) = 1.6885x(n) = 8.4424 + 16.8848(-1)^n$. Matlab script:

```
% P0318a: y(n) = sum_{m=0}^{3} x(n-2m)-sum_{l=1}^{3}
(0.81)^l y(n-2l)
% x(n) = 5+10(-1) ^ n;
clc; close all;
n = [0:50]; a = [1 0 0.81^2 0 0.81^4 0 0.81^6];
b = [1 0 1 0 1 0 1];
w = [0 pi]; A = [5 10]; theta = [0 0]; [H] =
freqresp(b,a,w);
magH = abs(H); phaH = angle(H); mag = A.*magH; pha =
phaH+theta;
term1 = w'*n; term2 = pha'*ones(1,length(n)); cos_term =
cos(term1+term2);
y1 = mag*cos_term;

% x = 5cos(n*w1+thetaq)+10cos(n*w2+theta2) where w1=0,
w2=pi,theta1=theta2=0;
w1=0;w2=pi;A1=5;A2=10;H1 = freqresp(b,a,w1);H2 =
freqresp(b,a,w2);
y11=A1*abs(H1)*cos(w1*n+angle(H1))+A2*abs(H2)*cos(w2*n+an
gle(H2));
dif_y = max(abs(y1-y11));

x = 5+10*(-1) .^ n; y2 = filter(b,a,x);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0318a');
subplot(2,1,1); Hs = stem(n,y1,'filled'); axis([-1 51 -10
30]);
set(Hs,'markersize',2); xlabel('n','FontSize',12);
ylabel('y(n)','FontSize',12);
title('Steady state response y_{ss}(n) for x(n) = 5+10(-
1)^{n}',...
'FontSize',12); ytick = [-10:5:25];
set(gca,'YTick',ytick);
subplot(2,1,2); Hs = stem(n,y2,'filled'); axis([-1 51 -10
30]);
set(Hs,'markersize',2); xlabel('n','FontSize',12);
```

```
ylabel('y(n)','FontSize',12);
title(['Output response y(n) using the filter function
for x(n) = ' ...
'5+10(-1)^{n}'],'FontSize',12); set(gca,'YTick',ytick);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0318a;
```

The steady-state responses are shown in Figure 3.43.

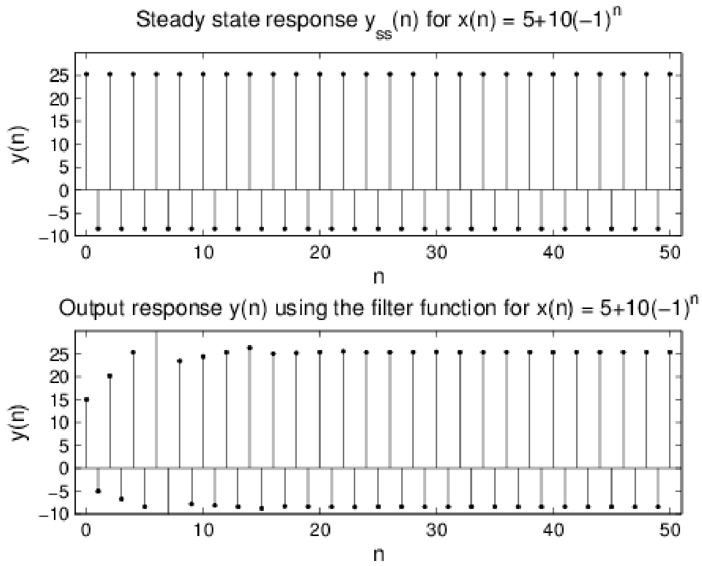


Figure 3.43: Steady-state response plots in Problem P3.18a

2. $x(n) = 1 + \cos(0.5\pi n + \pi/2)$: We need responses at $\omega = 0$ and $\omega = 0.5\pi$.

$$H\left(e^{j0}\right) = \frac{\sum_{m=0}^{3} e^{-j0}}{\sum_{\ell=0}^{3}(0.81)^{2\ell}e^{-j0}} = 1.6885 \text{ and } H\left(e^{j0.5\pi}\right) = \frac{\sum_{m=0}^{3} e^{-j\pi\ell}}{\sum_{\ell=0}^{3}(0.81)^{2\ell}e^{-j\pi\ell}} = 0$$

Hence the steady-state response is $y(n) = 1.6885$. Matlab script:

```
%% P0318b: y(n) = sum_{m=0}^{3} x(n-2m)-sum_{l=1}^{3}
(0.81)^l y(n-2l)
% x(n) = 1+cos(0.5*pi*n+pi/2);
clc; close all;
n = [0:50]; a = [1 0 0.81^2 0 0.81^4 0 0.81^6]; b = [1 0
1 0 1 0 1];
w = [0 pi/2]; A = [1 1]; theta = [0 pi/2]; [H] =
freqresp(b,a,w);
magH = abs(H); phaH = angle(H); mag = A.*magH; pha =
phaH+theta;
term1 = w'*n; term2 = pha'*ones(1,length(n)); cos_term =
cos(term1+term2);
y1 = mag*cos_term; x = 1+cos(0.5*pi*n+pi/2); y2 =
filter(b,a,x);
Hf_1 = figure;
```

```
set(Hf_1,'NumberTitle','off','Name','P0318b');
subplot(2,1,1); Hs = stem(n,y1,'filled'); axis([-1 51 0
2.5]);
ytick = [0:0.5:2.5]; set(Hs,'markersize',2);
set(gca,'YTick',ytick);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
title(['SS response y_{ss}(n): x(n) =
1+cos(0.5{\pi}n+\pi/2)'],'FontSize',12);
subplot(2,1,2); Hs = stem(n,y2,'filled');
set(Hs,'markersize',2);
axis([-1 51 0 2.5]); ytick = [0:0.5:2.5];
set(gca,'YTick',ytick);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
title(['Output response y(n) using the filter
function'],'FontSize',12);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0318b;
```

The steady-state responses are shown in Figure 3.44.

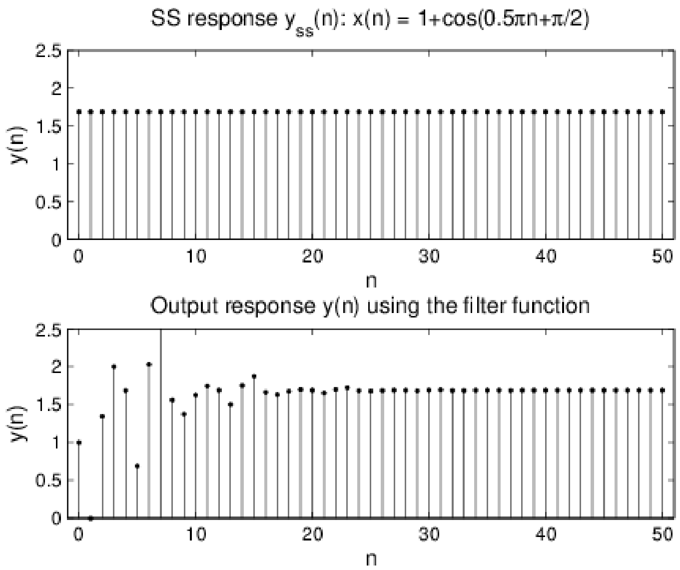


Figure 3.44: Steady-state response plots in Problem P3.18b

3. $x(n) = 2\sin(\pi n/4) + 3\cos(3\pi n/4)$: We need responses at $\omega = \pi/4$ and $\omega = 3\pi/4$.

$$H\left(e^{j0.25\pi}\right) = \frac{\sum_{m=0}^{3} e^{-j0.5\pi m}}{\sum_{\ell=0}^{3} (0.81)^{2\ell} e^{-j0.5\pi m}} = 0 \text{ and } H\left(e^{j0.75\pi}\right) = \frac{\sum_{m=0}^{3} e^{-j1.5\pi \ell}}{\sum_{\ell=0}^{3} (0.81)^{2\ell} e^{-j1.5\pi \ell}} = 0$$

Hence the steady-state response is $y(n) = 0$. Matlab script:

```
%% P0318c: y(n) = sum_{m=0}^{3} x(n-2m)-sum_{l=1}^{3}
(0.81)^l y(n-2l)
% x(n) = 2 * sin(\pi n/4)+3 * cos(3 \pi n/4);
clc; close all;
```

```
n = [0:50]; a = [1 0 0.81^2 0 0.81^4 0 0.81^6]; b = [1 0
1 0 1 0 1];
w = [pi/4 3*pi/4]; A = [2 3]; theta = [-pi/2 0]; [H] =
freqresp(b,a,w);
magH = abs(H); phaH = angle(H); mag = A.*magH; pha =
phaH+theta;
term1 = w'*n; term2 = pha'*ones(1,length(n)); cos_term =
cos(term1+term2);
y1 = mag*cos_term; x = 2*sin(pi*n/4)+3*cos(3*pi*n/4); y2
= filter(b,a,x);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0318c');
subplot(2,1,1);
Hs = stem(n,y1,'filled'); axis([-1 51 -3 4]);
set(Hs,'markersize',2);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
title(['SS response y_{ss}(n): x(n) =
2sin({\pi}n/4)+3cos(3 \pi n/4)'],...
'FontSize',12);
subplot(2,1,2); Hs = stem(n,y2,'filled'); axis([-1 51 -3
4]);
set(Hs,'markersize',2); xlabel('n','FontSize',12);
ylabel('y(n)',...
'FontSize',12);
title(['Output response y(n) using the filter
function'],'FontSize',12);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0318c;
```

The steady-state responses are shown in Figure 3.45.

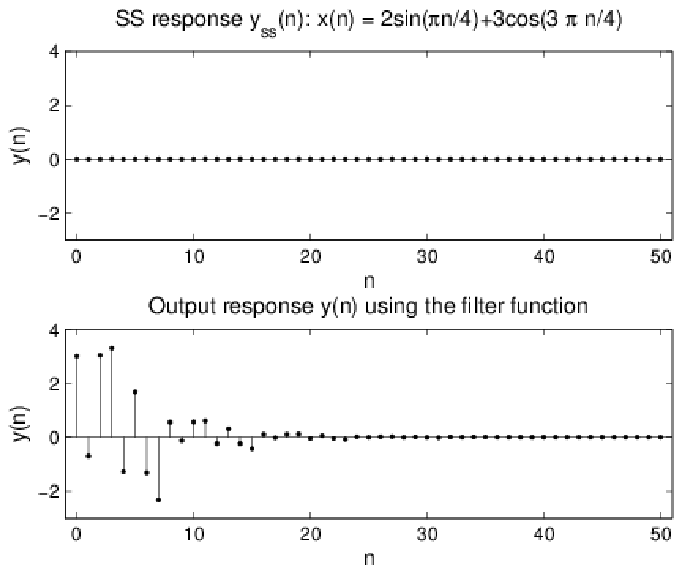


Figure 3.45: Steady-state response plots in Problem P3.18c

4. $x(n) = \sum_{k=0}^{5}(k + 1)\cos(\pi kn/4)$: We need responses at ω = $k\pi/4$, k = 0, 1, 2, 3, 4, 5.

$$H\left(e^{j0}\right) = \frac{\sum_{m=0}^{3} e^{-j0}}{\sum_{\ell=0}^{3}(0.81)^{2\ell}e^{-j0}} = 1.6885 = H\left(e^{j\pi}\right) \text{ and}$$
$$H\left(e^{j0.25\pi}\right) = H\left(e^{j0.5\pi}\right) = H\left(e^{j0.75\pi}\right) = H\left(e^{j1.25\pi}\right) = 0$$

Hence the steady-state response is $y(n) = 1.6885 + 8.4425\cos(n\pi)$. Matlab script:

```
%% P0318d: y(n) = sum_{m=0}^{3} x(n-2m)-sum_{l=1}^{3}
(0.81)^l y(n-2l)
% x(n) = sum_{k = 0}^{5} (k+1) cos(pi*k*n/4);
clc; close all;
n = [0:50]; a = [1 0 0.81^2 0 0.81^4 0 0.81^6]; b = [1 0
1 0 1 0 1];
k = [0:5]; w = pi /4*k; A = (k+1); theta =
zeros(1,length(k));
[H] = freqresp(b,a,w); magH = abs(H); phaH = angle(H);
mag = A.*magH;
pha = phaH+theta; term1 = w'*n; term2 =
pha'*ones(1,length(n)); cos_term = ...
cos(term1+term2); y1 = mag*cos_term; x = A*cos(term1); y2
= filter(b,a,x);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0318d');
subplot(2,1,1); Hs = stem(n,y1,'filled'); axis([-1 51 -10
15]);
set(Hs,'markersize',2); xlabel('n','FontSize',12);
title(['SS response y_{ss}(n): x(n) = sum_{0}^{5}
(k+1)cos({\pi}kn/4)'],...
'FontSize',12);
ytick = [-20:5:30]; set(gca,'YTick',ytick);
ylabel('y(n)','FontSize',12);
subplot(2,1,2); Hs = stem(n,y2,'filled'); axis([-1 51 -10
15]);
set(Hs,'markersize',2); xlabel('n','FontSize',12);
title(['Output response y(n) using the filter
function'],'FontSize',12);
ytick = [-20:5:30]; set(gca,'YTick',ytick);
ylabel('y(n)','FontSize',12);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0318d;
```

The steady-state responses are shown in Figure 3.46.

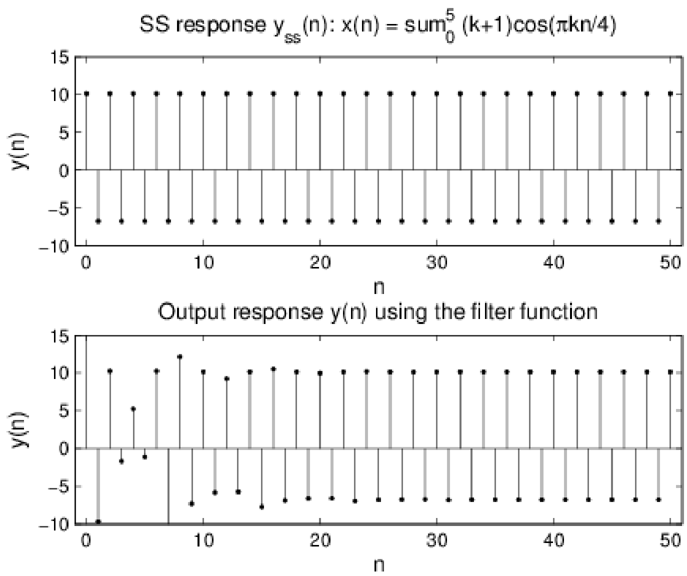


Figure 3.46: Steady-state response plots in Problem P3.18d

5. $x(n) = \cos(\pi n)$: We need response at $\omega = \pi$.

$$H\left(e^{j\pi}\right) = \frac{\sum_{m=0}^{3} e^{-j2\pi m}}{\sum_{\ell=0}^{3}(0.81)^{2\ell} e^{-j2\pi m}} = 1.6885$$

Hence the steady-state response is $y(n) = 1.6885 \cos(\pi n)$. Matlab script:

```
%% P0318e: y(n) = sum_{m=0}^{3} x(n-2m)-sum_{l=1}^{3}
(0.81)^l y(n-2l)
% x(n) = cos(pi*n);
clc; close all;
n = [0:50]; a = [1 0 0.81^2 0 0.81^4 0 0.81^6]; b = [1 0
1 0 1 0 1];
w = [pi]; A = [1]; theta = [0]; [H] = freqresp(b,a,w);
magH = abs(H);
phaH = angle(H); mag = A.*magH; pha = phaH+theta; term1 =
w'*n;
term2 = pha'*ones(1,length(n)); cos_term =
cos(term1+term2); y1 = mag*cos_term;
x = cos(pi*n); y2 = filter(b,a,x);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0318e');
subplot(2,1,1); Hs = stem(n,y1,'filled'); axis([-1 51 -2
2]);
set(Hs,'markersize',2); xlabel('n','FontSize',12);
title(['SS response y_{ss}(n) for x(n) = cos(\pi \times
n)'],'FontSize',12);
ytick = [-2:0.5:2]; set(gca,'YTick',ytick);
ylabel('y(n)','FontSize',12);
subplot(2,1,2); Hs = stem(n,y2,'filled'); axis([-1 51 -2
```

```
2]);
set(Hs,'markersize',2); xlabel('n','FontSize',12);
title(['Output response y(n) using the filter
function'],'FontSize',12);
ytick = [-2:0.5:2]; set(gca,'YTick',ytick);
ylabel('y(n)','FontSize',12);
set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0318e;
```

The steady-state responses are shown in Figure 3.47.

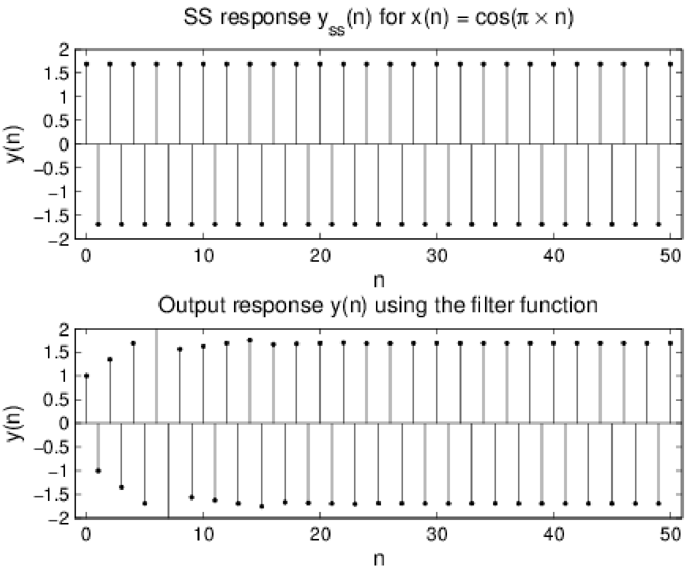


Figure 3.47: Steady-state response plots in Problem P3.18e

P3.19

An analog signal $x_a(t) = \sin(1000\pi t)$ is sampled using the following sampling intervals. In each case, plot the spectrum of the resulting discrete-time signal.

1. T_s = 0.1 ms
2. T_s = 1 ms
3. T_s = 0.01 sec

Solutions

1. T_s = 0.1 ms: Matlab script:

```
% P3.19
%% P0319a: x_a(t) = sin(1000*pi*t); T_s = 0.1 ms;
clc; close all;
%
Ts = 0.0001; n = [-250:250]; x = sin(1000*pi*n*Ts); w =
```

```
linspace(-pi,pi,501);
X = dtft(x,n,w); magX = abs(X); phaX = angle(X);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0319a');
subplot(2,1,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 0 300]);
wtick = [-1:0.2:1]; magtick = [0:100:300];
set(gca,'XTick',wtick);
xlabel('\omega / \pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response x_1(n) = sin(1000 \pi n T_s),
T_s = 0.1 msec'...
,'FontSize',12); set(gca,'YTick',magtick);
subplot(2,1,2); plot(w/pi,phaX*180/pi,'LineWidth',1.5);
axis([-1 1 -180 180]);
wtick = [-1:0.2:1]; magtick = [-180:60:180];
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase response x_1(n) = sin(1000 \pi n T_s), T_s =
0.1 msec'...
,'FontSize',12); set(gca,'XTick',wtick);
set(gca,'YTick',magtick);
print -deps2 ../EPSFILES/P0319a;
```

The spectra are shown in Figure 3.48.

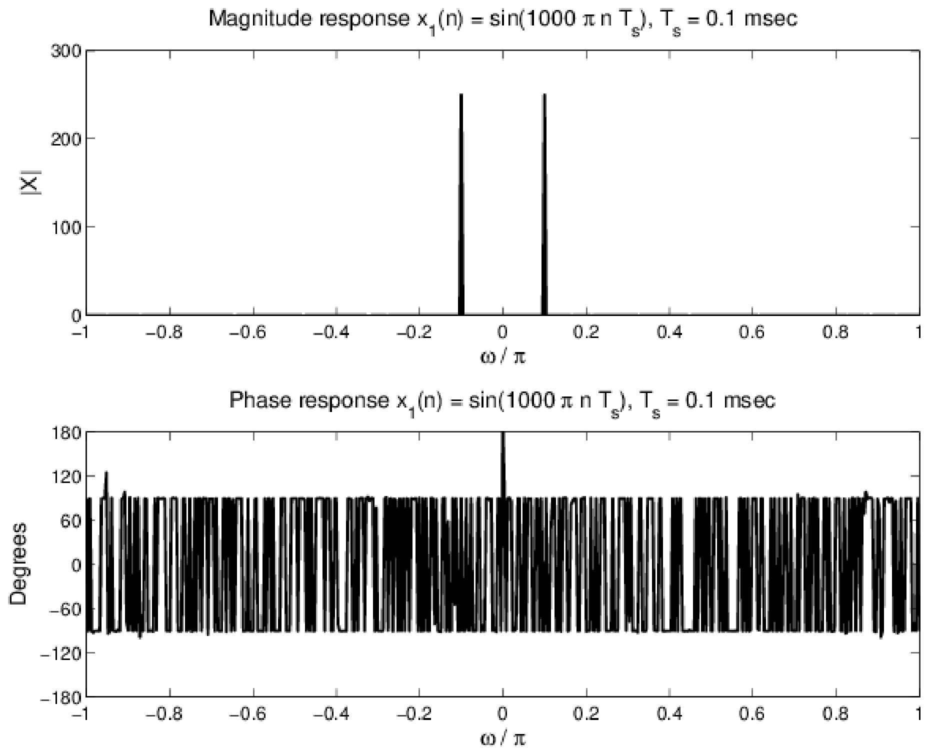


Figure 3.48: Spectrum plots in Problem P3.19a

2. T_s = 1 ms: Matlab script:

```
%% P0319b: x_a(t) = sin(1000*pi*t); T_s = 1 ms;
clc; close all;
%
Ts = 0.001; n = [-25:25]; x = sin(1000*pi*n*Ts); w = [-
500:500]*pi/500;
X = dtft(x,n,w); magX = abs(X); phaX = angle(X);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0319b');
subplot(2,1,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 -1 1]);
xlabel('\omega / \pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response x_1(n) = sin(1000 \pi n T_s),
T_s = 1 msec'...
,'FontSize',12);
wtick = [-1:0.2:1]; set(gca,'XTick',wtick);
subplot(2,1,2); plot(w/pi,phaX*180/pi,'LineWidth',1.5);
axis([-1 1 -180 180]);
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase response x_1(n) = sin(1000 \pi n T_s), T_s =
1 msec'...
,'FontSize',12); magtick = [-180:60:180];
wtick = [-1:0.2:1]; set(gca,'XTick',wtick);
set(gca,'YTick',magtick);
print -deps2 ../EPSFILES/P0319b;
```

The spectra are shown in Figure 3.49.

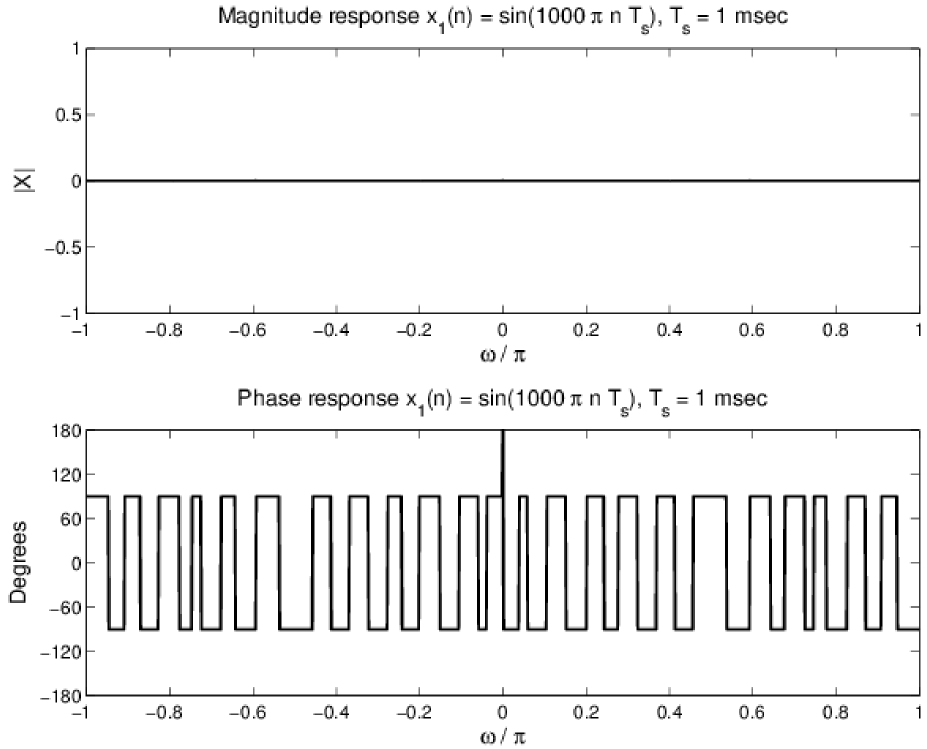


Figure 3.49: Spectrum plots in Problem P3.19b

3. T_s = 0.01 sec: Matlab script:

```
%% P0319c: x_a(t) = sin(1000*pi*t); T_s = 0.01 sec;
clc; close all;
%
Ts = 0.01; n = [-25:25]; x = sin(1000*pi*n*Ts); w = [-
500:500]*pi/500;
X = dtft(x,n,w); magX = abs(X); phaX = angle(X);
%
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0319c');
subplot(2,1,1); plot(w/pi,magX,'LineWidth',1.5); axis([-1
1 -1 1]);
xlabel('\omega / \pi','FontSize',12);
ylabel('|X|','FontSize',12);
title('Magnitude response x_1(n) = sin(1000 \pi n
T_s),T_s = 0.01 sec'...
,'FontSize',12); wtick = [-1:0.2:1];
set(gca,'XTick',wtick);
subplot(2,1,2); plot(w/pi,phaX*180/pi,'LineWidth',1.5);
axis([-1 1 -180 180]);
xlabel('\omega / \pi','FontSize',12);
ylabel('Degrees','FontSize',12);
title('Phase response x_1(n) = sin(1000 \pi n T_s),T_s =
```

```
0.01 sec'...
,'FontSize',12); wtick = [-1:0.2:1];
set(gca,'XTick',wtick);
magtick = [-180:60:180]; set(gca,'YTick',magtick);
print -deps2 ../EPSFILES/P0319c;
```

The spectra are shown in Figure 3.50.

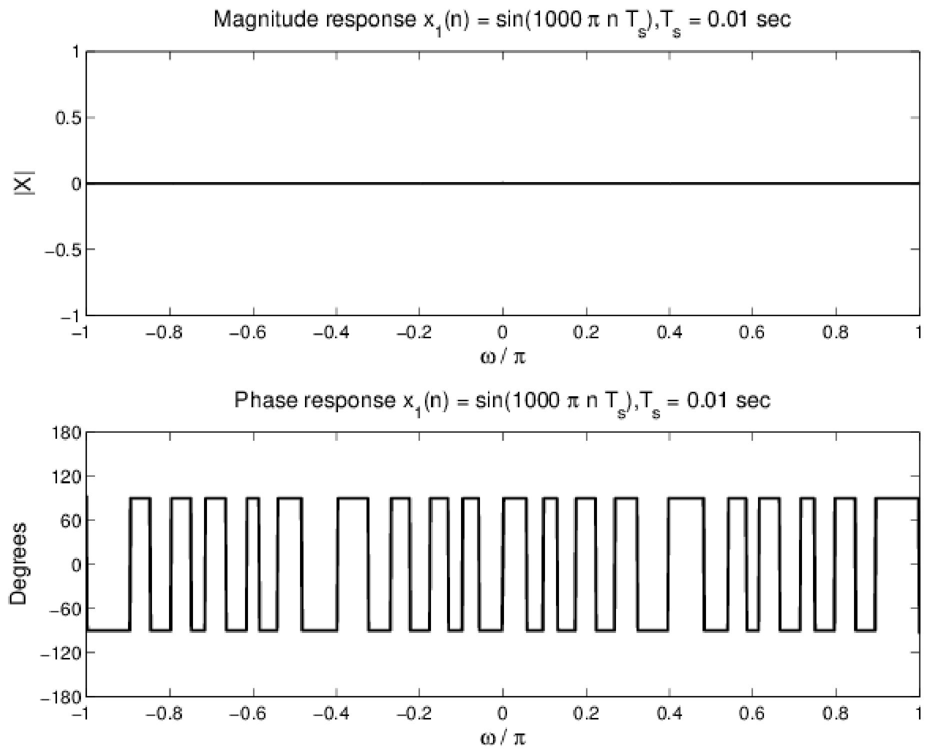


Figure 3.50: Spectrum plots in Problem P3.19c

P3.20

We implement the following analog filter using a discrete filter.

$$x_a(t) \longrightarrow \boxed{\text{A/D}} \xrightarrow{x(n)} \boxed{h(n)} \xrightarrow{y(n)} \boxed{\text{D/A}} \longrightarrow y_a(t)$$

The sampling rate in the A/D and D/A is 8000 sam/sec, and the impulse response is $h(n) = (-0.9)^n\ u(n)$.

1. What is the digital frequency in $x(n)$ if $x_a(t) = 10\cos(10{,}000\pi t)$?
2. Determine the steady-state output $y_a(t)$ if $x_a(t) = 10\cos(10{,}000\pi t)$.
3. Determine the steady-state output $y_a(t)$ if $x_a(t) = 5\sin(8{,}000\pi t)$.
4. Find two other analog signals $x_a(t)$, with different analog frequencies, that will give the same steady-state output $y_a(t)$ when $x_a(t) = 10\cos(10{,}000\pi t)$ is applied.
5. To prevent aliasing, a prefilter would be required to process $x_a(t)$ before it passes to the A/D converter. What type of filter should be used, and what should be the largest cutoff frequency that would work for the given configuration?

Solutions

(a) $x_a(t) = 10\cos(10000\pi t)$. Hence $x(n) = x_a(nT_s) = 10\cos(10000\pi n 0.000125) = 10\cos(1.25\pi n)$.
Therefore, the digital frequency is (1.25 - 2)π = -0.75π rad/sam.
(b) The steady-state response when $x(n) = 10\cos(-0.75\pi n) = 10\cos(0.75\pi n)$: The frequency response is

$$H\left(e^{j\omega}\right) = \mathcal{F}\left[h(n)\right] = \mathcal{F}\left[(-0.9)^n u(n)\right] = \frac{1}{1+0.9e^{j\omega}}.$$

At ω = -0.75π, the response is

$$H\left(e^{j0.75\pi}\right) = \frac{1}{1+0.9e^{j0.75\pi}} = 0.7329\left(\angle 1.0517^c\right).$$

Hence

$$y_{ss}(n) = 10\,(0.7329)\cos(0.75\pi n + 1.0517)$$

which after D/A conversion gives $y_{ss}(t)$ as

$$y_{ss,a}(t) = 7.329\cos(6000\pi t 1.0517).$$

(c) The steady-state DC gain is obtained by setting ω = 0 which is equal to $H\left(e^{j0}\right)$= 1/(1+0.9) = 0.5263.
Hence $y_{ss}(n) = 10\,(0.5263) = y_{ss,a}(t) = 5.263$.
(d) Aliased frequencies of F_0 for the given sampling rate F_s are $F_0 + kF_s$. Now for $F_0 = 5$ KHz and $F_s = 8$
KHz, the aliased frequencies are 5 + 8k = {13, 21, . . .} KHz. Therefore, two other $x_a(t)$'s are

10 cos(26000πt) and 10 cos(42000πt).

(e) The prefilter should be a lowpass filter with the cutoff frequency of 4 KHz.

P3.21

Consider an analog signal $x_a(t) = \cos(20\pi t)$, $0 \le t \le 1$. It is sampled at T_s= 0.01, 0.05, and 0.1 sec intervals to obtain $x(n)$.
1. For each T_s plot $x(n)$.
2. Reconstruct the analog signal $y_a(t)$ from the samples $x(n)$ using the sinc interpolation (use $\Delta t = 0.001$) and determine the frequency in $y_a(t)$ from your plot. (Ignore the end effects.)
3. Reconstruct the analog signal $y_a(t)$ from the samples $x(n)$ using the cubic spline interpolation, and determine the frequency in $y_a(t)$ from your plot. (Again, ignore the end effects.)
4. Comment on your results.

Solutions

1. Plots of $x(n)$ for each T_s. Matlab script:

```
% P2.21
```

```
%% P0321a: plot x(n) for T_s = 0.01 sec,0.05 sec,0.1 sec
% x_a(t) = cos(20*pi*t);
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0321a');
T_s1 = 0.01; n1 = [0:100]; x1 = cos(20*pi*n1*T_s1);
subplot(3,1,1); Hs = stem(n1,x1,'filled'); axis([-5 105 -
1.2 1.2]);
set(Hs,'markersize',2); xlabel('n','FontSize',12);
title(['x(n) = cos(20{\pi}nT_s) for T_s = 0.01
sec'],'FontSize',12);
ylabel('x(n)','FontSize',12);
T_s2 = 0.05; n2 = [0:20]; x2 = cos(20*pi*n2*T_s2);
subplot(3,1,2); Hs = stem(n2,x2,'filled');
set(Hs,'markersize',2);
set(gca,'XTick',[0:20]); axis([-2 22 -1.2 1.2]);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
title(['x(n) = cos(20{\pi}nT_s) for T_s = 0.05
sec'],'FontSize',12);
T_s3 = 0.1; n3 = [0:10]; x3 = cos(20*pi*n3*T_s3);
subplot(3,1,3); Hs = stem(n3,x3,'filled');
set(Hs,'markersize',2);
set(gca,'XTick',[0:10]); axis([-1 11 -1.2 1.2]);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
title(['x(n) = cos(20{\pi}nT_s) for T_s = 0.1
sec'],'FontSize',12);
print -deps2 ../EPSFILES/P0321a;
```

The plots are shown in Figure 3.51.

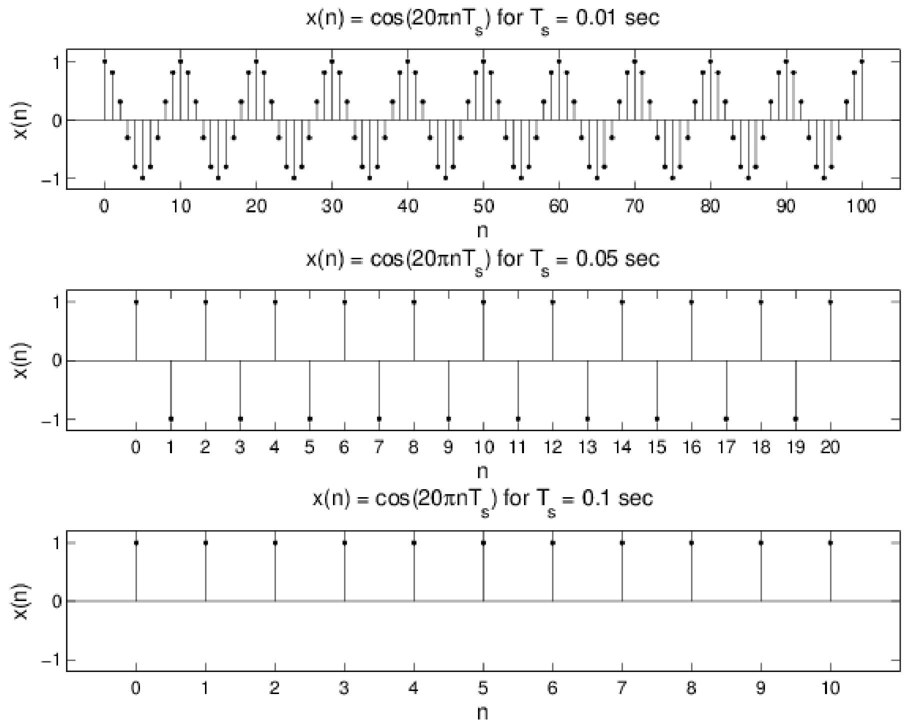


Figure 3.51: Plots of $x(n)$ for various T_s in Problem P3.21a.

2. Reconstruction from $x(n)$ using the sinc interpolation. Matlab script:

```
%% P0321b Sinc Interpolation: x_a(t) = cos(20*pi*t); 0 <=
t <= 1;
% T_s = 0.01 sec,0.05 sec and 0.1 sec;
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0321b');
%
Ts1 = 0.01; Fs1 = 1/Ts1; n1 = [0:100]; nTs1 = n1*Ts1;
x1 = cos(20*pi*nTs1); Dt = 0.001; t = 0:Dt:1; xa1 =
x1*sinc(Fs1*(ones(length(n1),1)*t-
nTs1'*ones(1,length(t))));
subplot(3,1,1); plot(t,xa1,'LineWidth',1.5); axis([0 1 -
1.2 1.2]);
xlabel('t in sec','FontSize',12);
ylabel('y_a(t)','FontSize',12);
title(['Sinc Interpolation: T_s = 0.01
sec'],'FontSize',12);grid;
%
Ts2 = 0.05; Fs2 = 1/Ts2; n2 = [0:20]; nTs2 = n2*Ts2;
x2 = cos(20*pi*nTs2); Dt = 0.001; t = 0:Dt:1; xa2 =
x2*sinc(Fs2*(ones(length(n2),1)*t-
nTs2'*ones(1,length(t))));
subplot(3,1,2); plot(t,xa2,'LineWidth',1.5); axis([0 1 -
```

```
1.2 1.2]);
xlabel('t in sec','FontSize',12);
ylabel('y_a(t)','FontSize',12);
title(['Sinc Interpolation: T_s = 0.05
sec'],'FontSize',12); grid;
%
Ts3 = 0.1; Fs3 = 1/Ts3; n3 = [0:10]; nTs3 = n3*Ts3; x3 =
cos(20*pi*nTs3);
Dt = 0.001; t = 0:Dt:1; xa3 =
x3*sinc(Fs3*(ones(length(n3),1)*t-
nTs3'*ones(1,length(t))));
subplot(3,1,3); plot(t,xa3,'LineWidth',1.5); axis([0 1 -
1.2 1.2]);
xlabel('t in sec','FontSize',12);
ylabel('y_a(t)','FontSize',12);
title(['Sinc Interpolation: T_s = 0.1
sec'],'FontSize',12); grid;
print -deps2 ../EPSFILES/P0321b;
```

The reconstruction is shown in Figure 3.52.

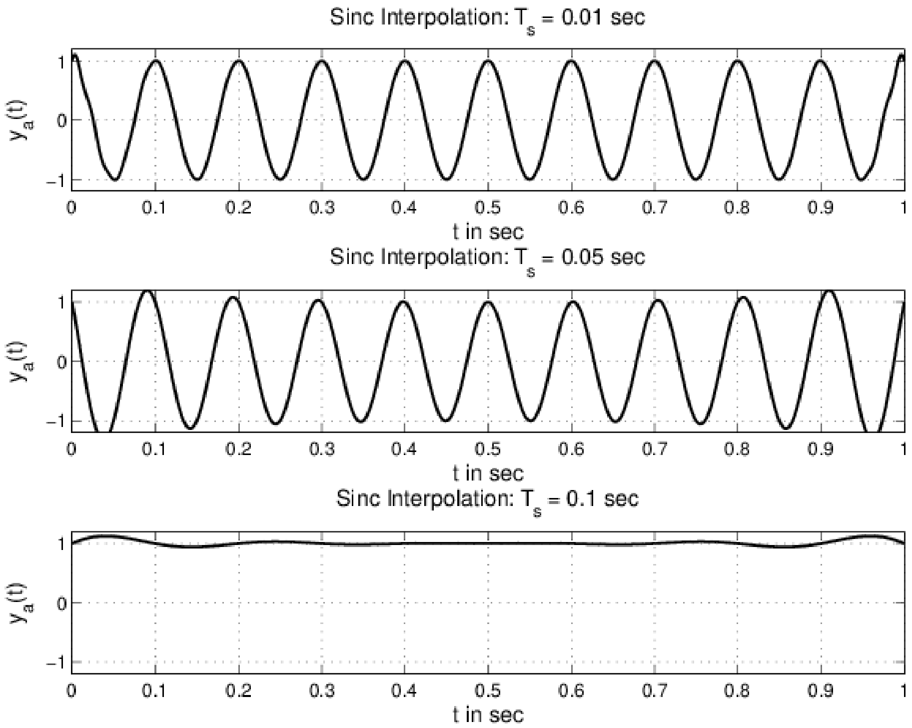


Figure 3.52: The sinc interpolation in Problem P3.21b.

3. Reconstruction from $x(n)$ using the spline interpolation. Matlab script:

```
%% P0321c Spline Interpolation: x_a(t) = cos(20*pi*t); 0
<= t <= 1;
% T_s = 0.01 sec,0.05 sec and 0.1 sec;
```

```
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0321c');
%
Ts1 = 0.01; Fs1 = 1/Ts1; n1 = [0:100]; nTs1 = n1*Ts1;
x1 = cos(20*pi*nTs1); Dt = 0.001; t = 0:Dt:1; xa1 =
spline(nTs1,x1,t);
subplot(3,1,1); plot(t,xa1,'LineWidth',1.5); axis([0 1 -
1.2 1.2]);
xlabel('t in sec','FontSize',12);
ylabel('y_a(t)','FontSize',12);
title(['Spline Interpolation: T_s = 0.01
sec'],'FontSize',12);grid;
%
Ts2 = 0.05; Fs2 = 1/Ts2; n2 = [0:20]; nTs2 = n2*Ts2;
x2 = cos(20*pi*nTs2); Dt = 0.001; t = 0:Dt:1; xa2 =
spline(nTs2,x2,t);
subplot(3,1,2); plot(t,xa2,'LineWidth',1.5); axis([0 1 -
1.2 1.2]);
xlabel('t in sec','FontSize',12);
ylabel('y_a(t)','FontSize',12);
title(['Spline Interpolation: T_s = 0.05
sec'],'FontSize',12); grid;
%
Ts3 = 0.1; Fs3 = 1/Ts3; n3 = [0:10]; nTs3 = n3*Ts3; x3 =
cos(20*pi*nTs3);
Dt = 0.001; t = 0:Dt:1; xa3 = spline(nTs3,x3,t);
subplot(3,1,3); plot(t,xa3,'LineWidth',1.5); axis([0 1 -
1.2 1.2]);
xlabel('t in sec','FontSize',12);
ylabel('y_a(t)','FontSize',12);
title(['Spline Interpolation: T_s = 0.1
sec'],'FontSize',12); grid;
print -deps2 ../EPSFILES/P0321c;
```

The reconstruction is shown in Figure 3.53.

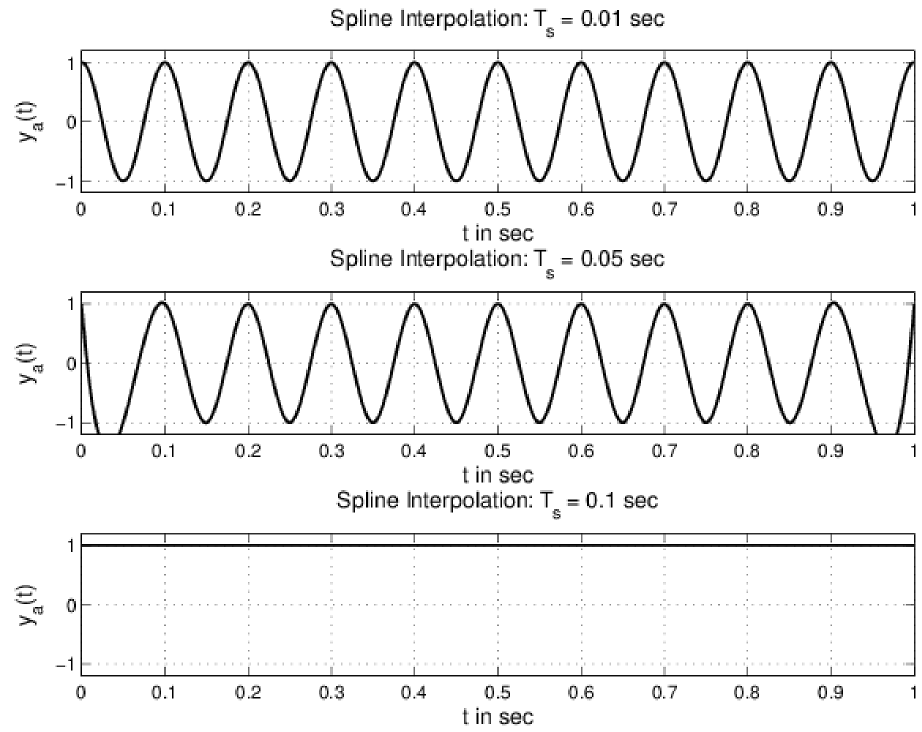


Figure 3.53: The sinc interpolation in Problem P3.21c.

4. Comments: From the plots in Figures it is clear that reconstructions from samples at T_s = 0.01 and 0.05 depict the original frequency (excluding end effects) but reconstructions for T_s = 0.1 show the original frequency aliased to zero. Furthermore, the cubic spline interpolation is a better reconstruction than the sinc interpolation, that is, the sinc interpolation is more susceptible to boundary effect.

P3.22

Consider the analog signal $x_a(t) = \cos(20\pi t + \theta)$, $0 \le t \le 1$. It is sampled at $T_s = 0.05$ sec intervals to obtain $x(n)$. Let $\theta = 0, \pi/6, \pi/4, \pi/3, \pi/2$. For each of these θ values, perform the following.

1. Plot $x_a(t)$ and superimpose $x(n)$ on it using the ***plot(n,x,'o')*** function.
2. Reconstruct the analog signal $y_a(t)$ from the samples $x(n)$ using the sinc interpolation (Use $\Delta t = 0.001$) and superimpose $x(n)$ on it.
3. Reconstruct the analog signal $y_a(t)$ from the samples $x(n)$ using the cubic spline interpolation and superimpose $x(n)$ on it.
4. You should observe that the resultant reconstruction in each case has the correct frequency but a different amplitude. Explain this observation. Comment on the role of phase of $x_a(t)$ on the sampling and reconstruction of signals.

Solutions

(a) Plots of $x_a(t)$ and $x(n)$ for $\theta = 0, \pi/6, \pi/4, \pi/3, \pi/2$. Matlab script:

```
% P3.22
%% P0322a: x_a(t) = cos(20*pi*t+theta); x(n) for theta =
0,pi/6,pi/4,pi/3, pi/2
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0322a');
Ts = 0.05; Fs = 1/Ts; Dt = 0.001; t = 0:Dt:1; n = [0:20];
nTs = n*Ts;
theta1 = 0; x_a1 = cos(20*pi*t+theta1); x1 =
cos(20*pi*nTs+theta1);
subplot(5,1,1); plot(t,x_a1,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x1,'o'); xlabel('t in sec','FontSize',12);
title('x_a(t) and x(n) for \theta =
0','FontSize',12);ylabel('Amplitude','FontSize',12);
theta2 = pi/6; x_a2 = cos(20*pi*t+theta2); x2 =
cos(20*pi*nTs+theta2);
subplot(5,1,2); plot(t,x_a2,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x2,'o'); xlabel('t in sec','FontSize',12);
title('x_a(t) and x(n) for \theta =
\pi/6','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta3 = pi/4; x_a3 = cos(20*pi*t+theta3); x3 =
cos(20*pi*nTs+theta3);
subplot(5,1,3); plot(t,x_a3,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x3,'o'); xlabel('t in sec','FontSize',12);
title('x_a(t) and x(n) for \theta =
\pi/4','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta4 = pi/3; x_a4 = cos(20*pi*t+theta4); x4 =
cos(20*pi*nTs+theta4);
subplot(5,1,4); plot(t,x_a4,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x4,'o'); xlabel('t in sec','FontSize',12);
title('x_a(t) and x(n) for \theta =
\pi/3','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta5 = pi/2; x_a5 = cos(20*pi*t+theta5); x5 =
```

```
cos(20*pi*nTs+theta5);
subplot(5,1,5); plot(t,x_a5,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x5,'o'); xlabel('t in sec','FontSize',12);
title('x_a(t) and x(n) for \theta =
\pi/2','FontSize',12);
ylabel('Amplitude','FontSize',12);
print -deps2 ../EPSFILES/P0322a;
```

The reconstruction is shown in Figure 3.54.

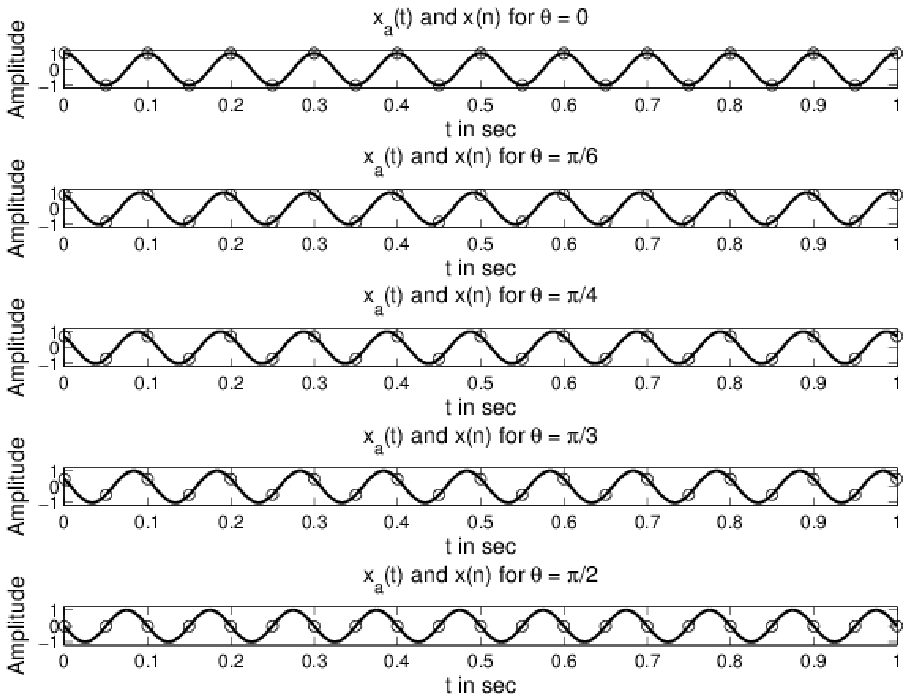


Figure 3.54: The sinc interpolation in Problem P3.22a.

(b) Reconstruction of the analog signal $y_a(t)$ from the samples $x(n)$ using the sinc interpolation (for $\theta = 0, \pi/6, \pi/4, \pi/3, \pi/2$. Matlab script:

```
%% P0322b: Sinc Interpolation for theta =
0,pi/6,pi/4,pi/3, pi/2
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0322b');
Ts = 0.05; Fs = 1/Ts; Dt = 0.001; t = 0:Dt:1; n = [0:20];
nTs = n*Ts;
theta1 = 0; x1 = cos(20*pi*nTs+theta1);
y_a1 = x1*sinc(Fs*(ones(length(n),1)*t-
nTs'*ones(1,length(t))));
subplot(5,1,1); plot(t,y_a1,'LineWidth',1.5); hold on;
plot(nTs,x1,'o'); axis([0 1 -1.2 1.2]); xlabel('t in
```

```
sec','FontSize',12);
title('Sinc Interpolation for \theta = 0','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta2 = pi/6; x2 = cos(20*pi*nTs+theta2);
y_a2 = x2*sinc(Fs*(ones(length(n),1)*t-
nTs'*ones(1,length(t))));
subplot(5,1,2); plot(t,y_a2,'LineWidth',1.5); hold on;
axis([0 1 -1.2 1.2])
plot(nTs,x2,'o'); xlabel('t in sec','FontSize',12);
title('Sinc Interpolation for \theta =
\pi/6','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta3 = pi/4; x3 = cos(20*pi*nTs+theta3);
y_a3 = x3*sinc(Fs*(ones(length(n),1)*t-
nTs'*ones(1,length(t))));
subplot(5,1,3); plot(t,y_a3,'LineWidth',1.5); hold on;
axis([0 1 -1.2 1.2])
plot(nTs,x3,'o'); xlabel('t in sec','FontSize',12);
title('Sinc Interpolation for \theta =
\pi/4','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta4 = pi/3; x4 = cos(20*pi*nTs+theta4);
y_a4 = x4*sinc(Fs*(ones(length(n),1)*t-
nTs'*ones(1,length(t))));
subplot(5,1,4); plot(t,y_a4,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x4,'o'); xlabel('t in sec','FontSize',12);
title('Sinc Interpolation for \theta =
\pi/3','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta5 = pi/2; x5 = cos(20*pi*nTs+theta5);
y_a5 = x5*sinc(Fs*(ones(length(n),1)*t-
nTs'*ones(1,length(t))));
subplot(5,1,5); plot(t,y_a5,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x5,'o'); xlabel('t in sec','FontSize',12);
title('Sinc Interpolation for \theta =
\pi/3','FontSize',12);
ylabel('Amplitude','FontSize',12);set(gcf,'paperpositionm
ode','auto');
print -deps2 ../EPSFILES/P0322b;
```

The reconstruction is shown in Figure 3.55.

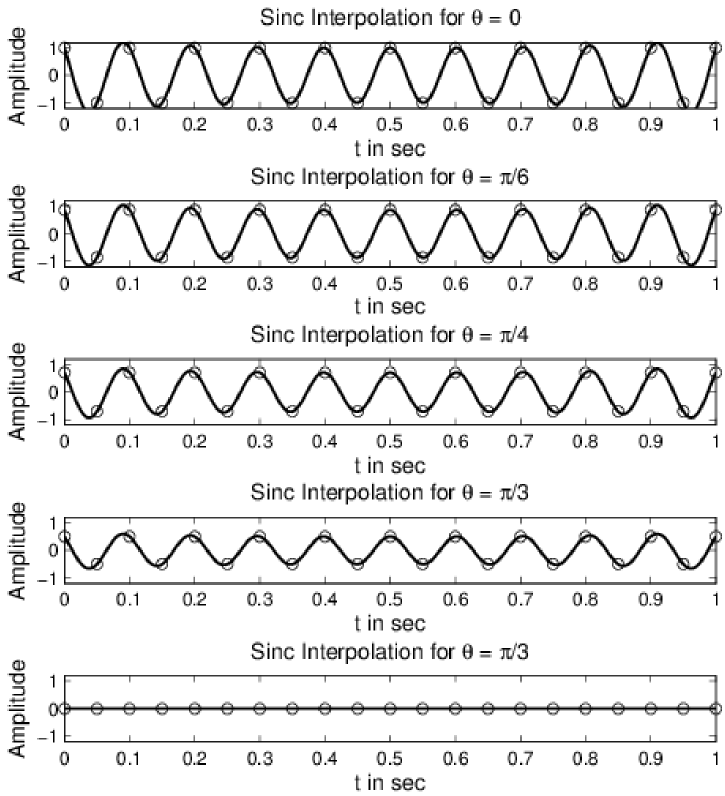


Figure 3.55: The sinc interpolation in Problem P3.22b.

(c) Reconstruction of the analog signal y_a (t) from the samples $x(n)$ using the spline interpolation (for $\theta = 0, \pi/6, \pi/4, \pi/3, \pi/2$. Matlab script:

```
%% P0322c: Spline Interpolation for theta =
0,pi/6,pi/4,pi/3, pi/2
clc; close all;
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0322c');
Ts = 0.05; Fs = 1/Ts; Dt = 0.001; t = 0:Dt:1; n = [0:20];
nTs = n*Ts;
theta1 = 0; x1 = cos(20*pi*nTs+theta1); y_a1 =
spline(nTs,x1,t);
subplot(5,1,1); plot(t,y_a1,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x1,'o'); xlabel('t in sec','FontSize',12);
title('Spline Interpolation for theta =
0','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta2 = pi/6; x2 = cos(20*pi*nTs+theta2); y_a2 =
spline(nTs,x2,t);
subplot(5,1,2); plot(t,y_a2,'LineWidth',1.5); hold on;
axis([0 1 -1.2 1.2]);
plot(nTs,x2,'o'); xlabel('t in sec','FontSize',12);
```

```
title('Spline Interpolation for theta =
\pi/6','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta3 = pi/4; x3 = cos(20*pi*nTs+theta3); y_a3 =
spline(nTs,x3,t);
subplot(5,1,3); plot(t,y_a3,'LineWidth',1.5); hold on;
axis([0 1 -1.2 1.2]);
plot(nTs,x3,'o'); xlabel('t in sec','FontSize',12);
title('Spline Interpolation for theta =
\pi/3','FontSize',12);
ylabel('Amplitude','FontSize',12);
theta4 = pi/3; x4 = cos(20*pi*nTs+theta4); y_a4 =
spline(nTs,x4,t);
subplot(5,1,4); plot(t,y_a4,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x4,'o'); ylabel('Amplitude','FontSize',12);
title('Spline Interpolation for theta =
\pi','FontSize',12);
xlabel('t in sec','FontSize',12);
theta5 = pi/2; x5 = cos(20*pi*nTs+theta5); y_a5 =
spline(nTs,x5,t);
subplot(5,1,5); plot(t,y_a5,'LineWidth',1.5); axis([0 1 -
1.2 1.2]); hold on;
plot(nTs,x5,'o'); ylabel('Amplitude','FontSize',12);
title('Spline Interpolation for theta =
\pi/2','FontSize',12);
xlabel('t in
sec','FontSize',12);set(gcf,'paperpositionmode','auto');
print -deps2 ../EPSFILES/P0322c;
```

The reconstruction is shown in Figure 3.56.

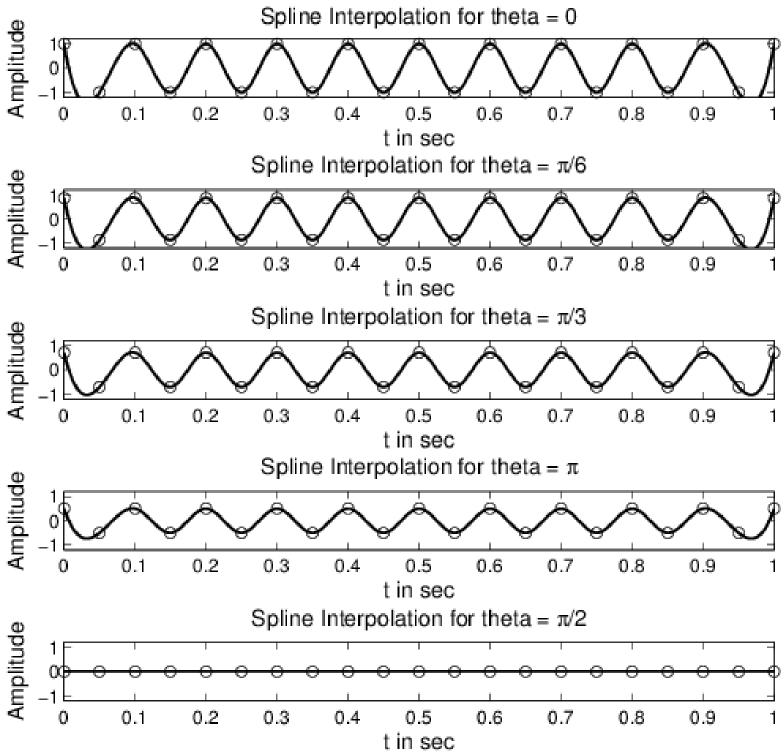


Figure 3.56: The sinc interpolation in Problem P3.22c.

(d) When a sinusoidal signal is sampled at $f = 2$ samples per cycle as is the case in this problem, then the resulting samples $x(n)$ has the amplitude that depends on the phase of the signal. In particular note that this amplitude is given by $\cos(\theta)$. Thus the amplitude of the reconstructed signal $y(t)$ is also equal to $\cos(\theta)$.

Chapter 4

P4.1

Determine the z-transform of the following sequences using the definition (4.1). Indicate the region of convergence for each sequence and verify the z-transform expression using MATLAB.

1. $x(n) = \{3, 2, 1, -2, -3\}$.
$\uparrow$
2. $x(n) = (0.8)^n u(n - 2)$. Verify the z-transform expression using MATLAB.
3. $x(n) = [(0.5)^n + (-0.8)^n]u(n)$. Verify the z-transform expression using MATLAB.
4. $x(n) = 2^n \cos(0.4\pi n)u(-n)$.
5. $x(n) = (n + 1)(3)^n u(n)$. Verify the z-transform expression using MATLAB.

Solutions

1. $x(n) = \{3, 2, 1, -2, -3\}$: Then $X(z) = 3z^2 + 2z + 1 - 2z^{-1} + 3z^{-2}$, $0 < |z| < \infty$.
$\uparrow$

Matlab verification:

```
% P4.1
%% P0401a.m
clc; close all;
b1 = [0 2 3]; a1 = [1]; [delta,n] = impseq(0,0,4);
xb1 = filter(b1,a1,delta); xb1 = fliplr(xb1); n1 = -
fliplr(n);
% the z-transform of x(-n) is X(1/z), considering the z-
transform of x(n) is X(z).
b2 = [1 -2 -3]; a2 = [1]; xb2 = filter(b2,a2,delta); n2 =
n;
[xa1,na1] = sigadd(xb1,n1,xb2,n2); xa2 = [0 0 3 2 1 -2 -3
0 0];
error = max(abs(xa1-xa2))
error =
      0
```

2. $x(n) = (0.8)^n u(n - 2)$: Then

$$X(z) = \sum_{n=2}^{\infty}(0.8)^n z^{-n} = (0.8)^2 z^{-2}\sum_{n=0}^{\infty}(0.8)^n z^{-n} = \frac{0.64z^{-2}}{1 - 0.8z^{-1}};\ |z| > 0.8$$

Matlab verification:

```
%% P0401b.m
clc; close all;
b = [0 0 0.64]; a = [1 -0.8]; [delta,n] = impseq(0,0,10);
xb1 = filter(b,a,delta);
```

```
[u,n] = stepseq(2,0,10); xb2 = ((0.8).^n).*u;
error = max(abs(xb1-xb2))
error =
   1.1102e-16
```

3. $x(n) = [(0.5)^n + (-0.8)^n]u(n)$: Then

$$X(z) = \mathcal{Z}\left[(0.5)^n u(n)\right] + \mathcal{Z}\left[(-0.8)^n u(n)\right] = \frac{1}{1-0.5z^{-1}} + \frac{1}{1+0.8z^{-1}};\ \{|z| > 0.5\} \cap \{|z| > 0.8\}$$

$$= \frac{2+0.3z^{-1}}{1+0.3z^{-1}-0.4z^{-2}};\ |z| > 0.8$$

Matlab verification:

```
%% P0401c.m
clc; close all;
b = [ 2 0.3]; a = [1 0.3 -0.4]; [delta,n] =
impseq(0,0,7);
xc1 = filter(b,a,delta);
[u,n] = stepseq(0,0,7); xc2 = (((0.5).^n).*u)+(((-
0.8).^n).*u);
error = max(abs(xc1-xc2))
error =
  1.1102e-16
```

4. $x(n) = 2^n \cos(0.4\pi n)u(-n)$: Consider

$$X(z) = \sum_{n=-\infty}^{\infty} 2^n \cos(0.4\pi n)u(-n)z^{-n} = \sum_{n=-\infty}^{0} 2^n \cos(0.4\pi n)z^{-n} = \sum_{n=0}^{\infty} 2^{-n} \cos(0.4\pi n)z^{n}$$

$$= \sum_{n=0}^{\infty} 2^{-n}\left(\frac{e^{j0.4\pi n}+e^{-j0.4\pi n}}{2}\right)z^n = \frac{1}{2}\sum_{n=0}^{\infty}\left(2^{-1}e^{j0.4\pi}z\right)^n + \frac{1}{2}\sum_{n=0}^{\infty}\left(2^{-1}e^{-j0.4\pi}z\right)^n$$

$$= \frac{1}{2}\left(\frac{1}{1-\frac{1}{2}e^{j0.4\pi}z}\right) + \frac{1}{2}\left(\frac{1}{1-\frac{1}{2}e^{-j0.4\pi}z}\right);\ |z| < 2$$

$$= \frac{1-[0.5\cos(0.4\pi)]z}{1-[\cos(0.4\pi)]z+0.25z^2};\ |z| < 2$$

Hence

$$X(z) = \frac{1-[0.5\cos(0.4\pi)]z}{1-2[0.5\cos(0.4\pi)]z+0.25z^2} = \frac{1-[0.5\cos(0.4\pi)]z}{1-[\cos(0.4\pi)]z+0.25z^2};\ |z| < 2$$

Matlab verification:

```
clc;close all;
b = [1 -0.5*cos(0.4*pi) 0];
a = [1 -cos(0.4*pi) 0.25];
[delta,n] = impseq(0,0,10);xd1 = filter(b,a,delta);
xd1 = fliplr(xd1);n = -fliplr(n);
[u,n2] = stepseq(-10,-10,0);
xd2 = 2.^n2.*cos(0.4*pi*n).*u;
error = max(abs(xd1-xd2))
```

```
error =
   2.7756e-17
```

5. $x(n) = (n + 1)(3)_n u(n)$: Consider

$$x(n) = (n + 1)(3)^n u(n) = n3^n u(n) + 3^n u(n)$$

Hence

$$X(z) = \mathcal{Z}\left[n3^n u(n)\right] + \mathcal{Z}\left[3^n u(n)\right] = -z\frac{d}{dz}\left(\frac{1}{1-3z^{-1}}\right) + \frac{1}{1-3z^{-1}};\ |z| > 3$$
$$= \frac{3z^{-1}}{1-6z^{-1}+9z^{-2}} + \frac{1}{1-3z^{-1}} = \frac{1-3z^{-1}}{1-9z^{-1}+27z^{-2}-27z^{-3}};\ |z| > 3$$

Matlab verification:

```
clc; close all;
b = [1 -3]; a = [1 -9 27 -27]; [delta,n1] =
impseq(0,0,7);
xb1 = filter(b,a,delta);
[u,n2] = stepseq(0,0,7); xb2 = ((n2+1).*(3.^n2)).*u;
error = max(abs(xb1-xb2))
error =
         0
```

P4.2

Consider the sequence $x(n) = (0.9)^n \cos(\pi n/4)u(n)$. Let

$$y(n) = \begin{cases} x(n/2), & n = 0, \pm 2, \pm 4, \cdots; \\ 0, & \text{otherwise.} \end{cases}$$

1. Show that the z-transform $Y(z)$ of $y(n)$ can be expressed in terms of the z-transform $X(z)$ of $x(n)$ as $Y(z) = X(z^2)$.
2. Determine $Y(z)$.
3. Using MATLAB, verify that the sequence $y(n)$ has the z-transform $Y(z)$.

Solutions

Consider the sequence $x(n) = (0.9)_n \cos(\pi n/4)u(n)$. Let

$$y(n) = \begin{cases} x(n/2), & n = 0, \pm 2, \pm 4, \cdots; \\ 0, & \text{otherwise.} \end{cases}$$

1. The z-transform $Y(z)$ of $y(n)$ in terms of the z-transform $X(z)$ of $x(n)$: Consider

$$Y(z) = \sum_{n=-\infty}^{\infty} y(n)z^{-n} = \sum_{n=-\infty}^{\infty} x(n/2)z^{-n};\ n = 0, \pm 1, \pm 2, \ldots$$
$$= \sum_{m=-\infty}^{\infty} x(m)z^{-2m} = \sum_{m=-\infty}^{\infty} x(m)\left(z^2\right)^{-m} = X(z^2)$$

2. The z-transform of $x(n)$ is given by

$$X(z) = \mathcal{Z}\left[(0.9)^n \cos(\pi n/4)u(n)\right] = \frac{1 - [(0.9)\cos(\pi/4)]z^{-1}}{1 - 2[(0.9)\cos(\pi/4)]z^{-1} + (0.9)^2 z^{-2}}$$

$$= \frac{1 - 0.6364z^{-1}}{1 - 1.2728z^{-1} + 0.81z^{-2}}; \ |z| > 0.9$$

Hence

$$Y(z) = \frac{1 - 0.6364z^{-2}}{1 - 1.2728z^{-2} + 0.81z^{-4}}; \ |z| > \sqrt{0.9} = 0.9487$$

3. Matlab verification:

```
% P4.2
clc; close all;
b = [1 0 -0.9*cos(pi/4)]; a = [1 0 2*-0.9*cos(pi/4) 0
0.81];
[delta,n1] = impseq(0,0,13); xb1 = filter(b,a,delta);
[u,n2] = stepseq(0,0,6); x1 =
(((0.9).^n2).*cos(pi*n2/4)).*u;
xb2 = zeros(1,2*length(x1)); xb2(1:2:end) = x1;
error = max(abs(xb1-xb2))
error =
   1.2442e-16
```

P4.3

Determine the z-transform of the following sequences using the z-transform table and the z-transform properties. Express $X(z)$ as a rational function in z^{-1}. Verify your results using MATLAB. Indicate the region of convergence in each case, and provide a pole-zero plot.

1. $x(n) = 2\delta(n - 2) + 3u(n - 3)$
2. $x(n) = 3(0.75)^n \cos(0.3\pi n)u(n) + 4(0.75)^n \sin(0.3\pi n)u(n)$
3. $x(n) = n\sin(\frac{\pi n}{3})u(n) + (0.9)^n u(n - 2)$
4. $x(n) = n^2 (2/3)^{n-2} u(n - 1)$
5. $x(n) = (n - 3)(\frac{1}{4})^{n-2} \cos\{\frac{\pi}{2}(n - 1)\}u(n)$

Solutions

1. $x(n) = 2\delta(n - 2) + 3u(n - 3)$:

$$X(z) = 2z^{-2} + 3z^{-3}\frac{1}{1 - z^{-1}} = \frac{2z^{-2} + z^{-3}}{1 - z^{-1}}; \ |z| > 1$$

Matlab script:

```
% P4.3
```

```
%% P0403a.m
clc; close all;
b = [0 0 2 1]; a = [1 -1];
[delta,n1] = impseq(0,0,9); xa1 = filter(b,a,delta);
[u,n2] = stepseq(3,0,9);[delta1,n3]=impseq(2,0,9);xa2 =
2*delta1+3*u;
error = max(abs(xa1-xa2))
Hf_1 =
figure;set(Hf_1,'NumberTitle','off','Name','P0403a');
[Hz,Hp,H] =
zplane(b,a);set(Hz,'linewidth',1);set(Hp,'linewidth',1);
title('Pole-Zero plot','FontSize',12);
print -deps2 ../EPSFILES/P0403a;
```

The pole-zero plot is shown in Figure 4.1.

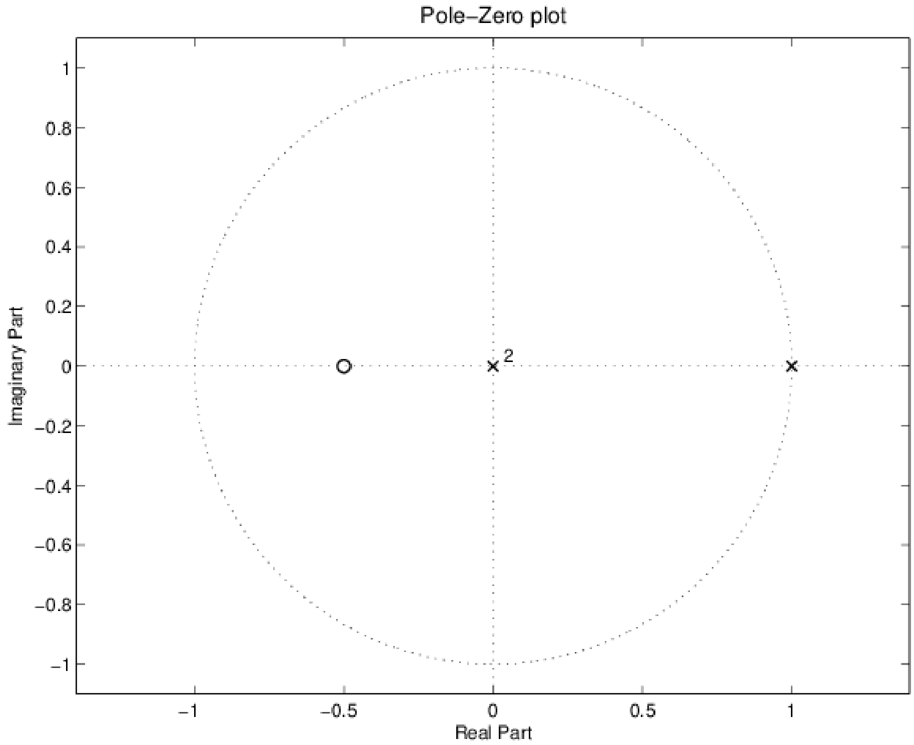


Figure 4.1: Problem P4.3.1 pole-zero plot

2. $x(n) = 3(0.75)^n \cos(0.3\pi n)u(n) + 4(0.75)^n \sin(0.3\pi n)u(n)$:

$$\begin{aligned} Z(z) &= 3\frac{1-[0.75\cos(0.3\pi)]z^{-1}}{1-2[0.75\cos(0.3\pi)]z^{-1}+(0.75)^2z^{-2}} + 4\frac{[0.75\sin(0.3\pi)]z^{-1}}{1-2[0.75\cos(0.3\pi)]z^{-1}+(0.75)^2z^{-2}} \\ &= \frac{3+[3\sin(0.3\pi)-2.25\cos(0,3\pi)]z^{-1}}{1-1.5\cos(0,3\pi)z^{-1}+0.5625z^{-2}} = \frac{3+1.1045z^{-1}}{1-0.8817z^{-1}+0.5625z^{-2}}; \ |z|>0.75 \end{aligned}$$

Matlab script:

```
%% P0403b.m
clc; close all;
```

```
b = [3 (3*sin(0.3*pi)-2.25*cos(0.3*pi))]; a = [1 -
1.5*cos(0.3*pi) 0.5625];
[delta,n1] = impseq(0,0,7); xb1 = filter(b,a,delta);
[u,n2] = stepseq(0,0, 7);
xb2 =
3*(((0.75).^n2).*cos(0.3*pi*n2)).*u+4*(((0.75).^n2).*sin(
0.3*pi*n2)).*u ;
error = max(abs(xb1-xb2))
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0403b');
[Hz,Hp,Hl] = zplane(b,a); set(Hz,'linewidth',1);
set(Hp,'linewidth',1);
title('Pole-Zero plot','FontSize',12);
print -deps2 ../epsfiles/P0403b;
```

The pole-zero plot is shown in Figure 4.2.

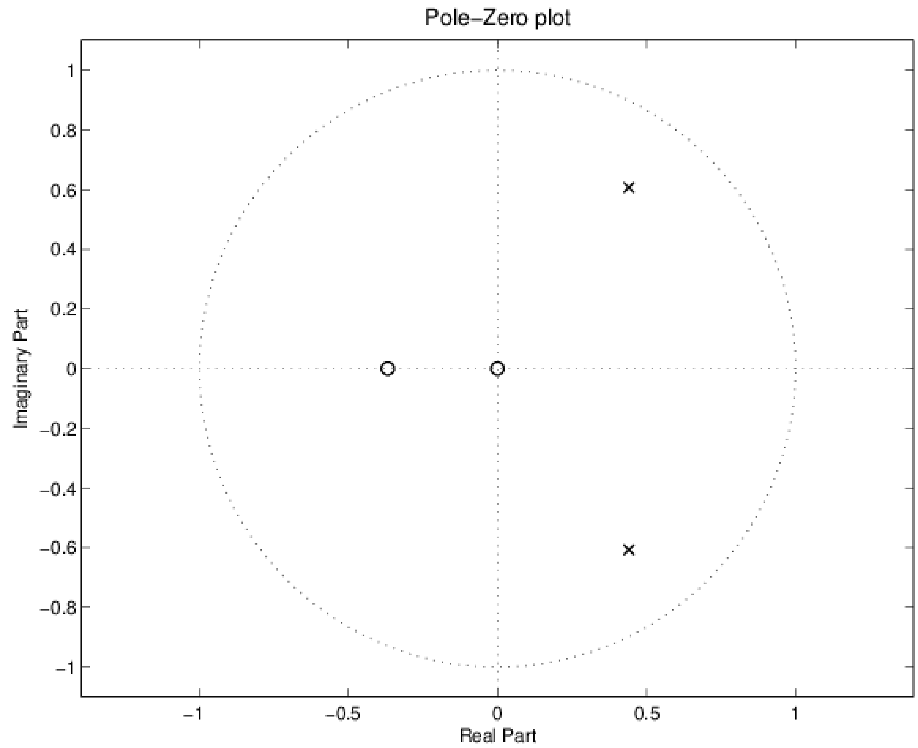


Figure 4.2: Problem P4.3.2 pole-zero plot

3. $x(n) = n\sin(\frac{\pi n}{3})u(n) + (0.9)^n u(n-2)$: Consider

$$\begin{aligned} x(n) &= n\sin\left(\frac{\pi n}{3}\right)u(n) + (0.9)^n u(n-2) = n\sin\left(\frac{\pi n}{3}\right)u(n) + 0.9^2(0.9)^{n-2}u(n-2) \\ &= n\sin\left(\frac{\pi n}{3}\right)u(n) + 0.81(0.9)^{n-2}u(n-2) \end{aligned}$$

Hence

$$
\begin{aligned}
X(z) &= \mathcal{Z}\left[n\sin\left(\frac{\pi n}{3}\right)u(n)\right] + \mathcal{Z}\left[0.81(0.9)^{n-2}u(n-2)\right] \\
&= -z\frac{d}{dz}\left(\mathcal{Z}\left[\sin\left(\frac{\pi n}{3}\right)u(n)\right]\right) + 0.81z^{-2}\mathcal{Z}\left[(0.9)^n u(n)\right] \\
&= -z\frac{d}{dz}\left(\frac{\sin(\pi/3)z^{-1}}{1-z^{-1}+z^{-2}}\right) + \frac{0.81z^{-2}}{1-0.9z^{-1}};\ |z|>1 \\
&= \frac{\sin(\pi/3)z^{-1}-\sin(\pi/3)z^{-3}}{1-2z^{-1}+3z^{-2}-2z^{-3}+z^{-4}} + \frac{0.81z^{-2}}{1-0.9z^{-1}};\ |z|>1 \\
&= \frac{0.866z^{-1}+0.0306z^{-2}-2.486z^{-3}+3.2094z^{-4}-1.62z^{-5}+0.81z^{-6}}{1-2.9z^{-1}+4.8z^{-2}-4.7z^{-3}+2.8z^{-4}-0.9z^{-5}};\ |z|>1
\end{aligned}
$$

Matlab script:

```
%% P0403c.m
clc; close all;
b = [0 sin(pi/3) (0.81-0.9*sin(pi/3)) -
(1.62+sin(pi/3)) ...
(0.9*sin(pi/3)+2.43) -1.62 0.81];
a = [1 -2.9 4.8 -4.7 2.8 -0.9]; [delta,n1] =
impseq(0,0,9);
xb1 = filter(b,a,delta);
[u2,n2] = stepseq(0,0,9); [u3,n3] = stepseq(2,0,9);
xb2 = (n2.*sin(pi/3*n2)).*u2+((0.9).^n3).*u3; error =
max(abs(xb1-xb2))
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0403c');
[Hz,Hp,Hl] = zplane(b,a); set(Hz,'linewidth',1);
set(Hp,'linewidth',1);
title('Pole-Zero plot','FontSize',12);
print -deps2 ../epsfiles/P0403c;
error =
   2.1039e-14
```

The pole-zero plot is shown in Figure 4.3.

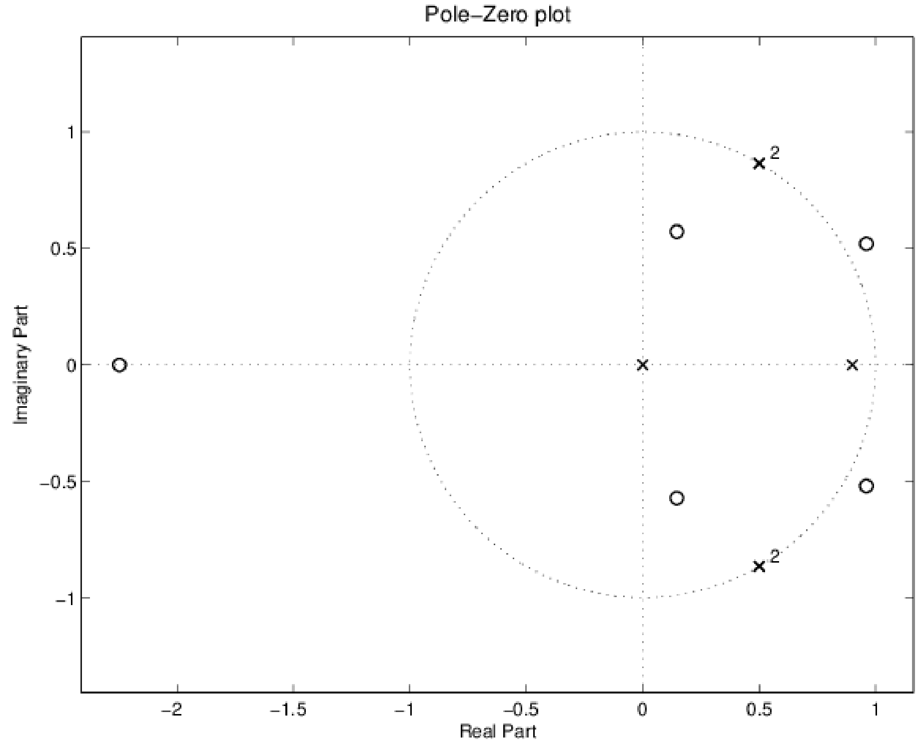


Figure 4.3: Problem P4.3.3 pole-zero plot

4. $x(n) = n^2(2/3)^{n-2}u(n-1)$: Consider

$$x(n) = n^2(2/3)^{n-2}u(n-1) = n^2(2/3)^{-1}(2/3)^{(n-1)}u(n-1) = \frac{3}{2}\left(n\left[n\left\{\left(\frac{2}{3}\right)^{(n-1)}u(n-1)\right\}\right]\right)$$

Let

$$x_1(n) = \left(\frac{2}{3}\right)^{(n-1)} u(n-1) \Rightarrow X_1(z) = \frac{z^{-1}}{1-\frac{2}{3}z^{-1}};\ |z| > \frac{2}{3}$$

Let

$$x_2(n) = n\,x_1(n) \Rightarrow X_2(z) = -z\frac{d}{dz}X_1(z) = \frac{z^{-1}}{1-\frac{4}{3}z^{-1}+\frac{4}{9}z^{-2}};\ |z| > \frac{2}{3}$$

Let

$$x_3(n) = n\,x_2(n) \Rightarrow X_3(z) = -z\frac{d}{dz}X_2(z) = \frac{z^{-1}-\frac{4}{9}z^{-3}}{1-\frac{8}{3}z^{-1}+\frac{8}{3}z^{-2}-\frac{32}{27}z^{-3}+\frac{16}{81}z^{-4}};\ |z| > \frac{2}{3}$$

Finally, $x(n) = \frac{3}{2}x_3(n)$. Hence

$$X(z) = \frac{3}{2}\left(\frac{z^{-1}-\frac{4}{9}z^{-3}}{1-\frac{8}{3}z^{-1}+\frac{8}{3}z^{-2}-\frac{32}{27}z^{-3}+\frac{16}{81}z^{-4}}\right) = \frac{\frac{3}{2}z^{-1}-\frac{2}{3}z^{-3}}{1-\frac{8}{3}z^{-1}+\frac{8}{3}z^{-2}-\frac{32}{27}z^{-3}+\frac{16}{81}z^{-4}};\ |z| > \frac{2}{3}$$

Matlab script:

```
%% P0403d.m
```

```
clc; close all; b = 3/2*[0 1 0 -4/9]; a = [1 -8/3 8/3 -
32/27 16/81];
[delta,n1] = impseq(0,0,8); xb1 = filter(b,a,delta);
[u,n2] = stepseq(1,0,8); xb2 = ((n2.^2).*((2/3).^(n2-
2))).*u;
error = max(abs(xb1-xb2))
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0403d');
[Hz,Hp,Hl] = zplane(b,a); set(Hz,'linewidth',1);
set(Hp,'linewidth',1);
title('Pole-Zero plot','FontSize',12);
print -deps2 ../epsfiles/P0403d;
error =
   9.7700e-15
```

The pole-zero plot is shown in Figure 4.4.

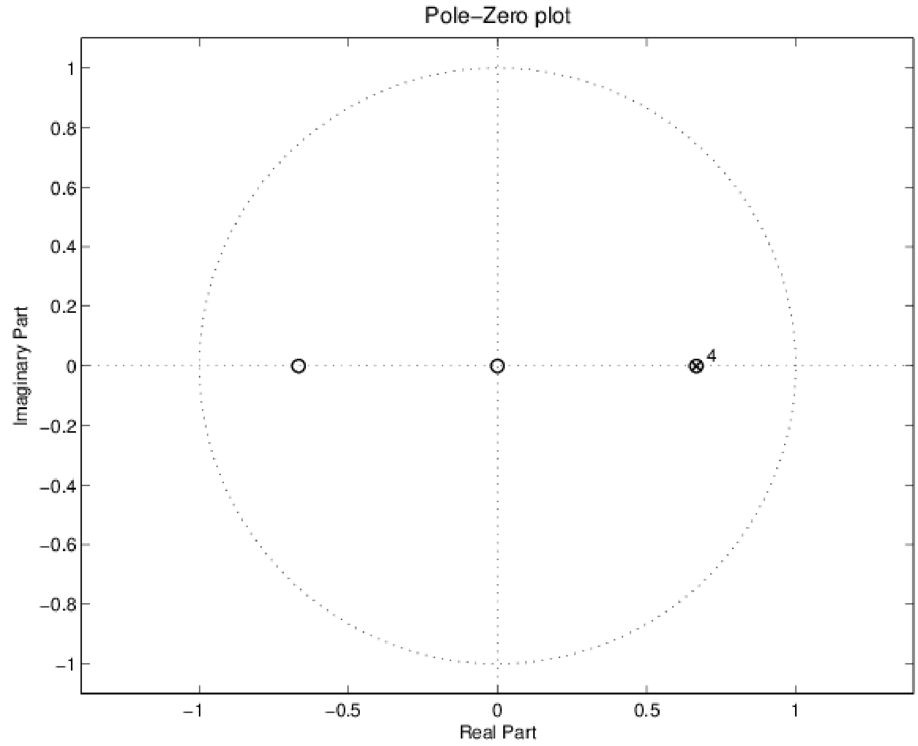


Figure 4.4: Problem P4.3.4 pole-zero plot

5. $x(n) = (n-3)\left(\frac{1}{4}\right)^{n-2}\cos\left\{\frac{\pi}{2}(n-1)\right\}u(n)$: Consider

$$x(n) = (n-3)\left(\frac{1}{4}\right)^{n-2}\cos\left\{\frac{\pi}{2}(n-1)\right\}u(n) = (n-3)\left(\frac{1}{4}\right)^{n-2}\sin\left\{\frac{\pi}{2}n\right\}u(n)$$

$$= (n-3)\left(\frac{1}{4}\right)^{-2}\left(\frac{1}{4}\right)^{n}\sin\left\{\frac{\pi}{2}n\right\}u(n) = 16(n-3)\left(\frac{1}{4}\right)^{n}\sin\left\{\frac{\pi}{2}n\right\}u(n)$$

$$= 16\,n\left(\frac{1}{4}\right)^{n}\sin\left\{\frac{\pi}{2}n\right\}u(n) - 48\left(\frac{1}{4}\right)^{n}\sin\left\{\frac{\pi}{2}n\right\}u(n)$$

Hence

$$
\begin{aligned}
X(z) &= \mathcal{Z}\left[16n\left(\frac{1}{4}\right)^n \sin\left\{\frac{\pi}{2}n\right\}u(n)\right] - \mathcal{Z}\left[48\left(\frac{1}{4}\right)^n \sin\left\{\frac{\pi}{2}n\right\}u(n)\right] \\
&= -z\frac{d}{dz}\left(\mathcal{Z}\left[\left(\frac{1}{4}\right)^n \sin\left\{\frac{\pi}{2}n\right\}u(n)\right]\right) - 48\frac{[(\frac{1}{4})\sin(\pi/2)]z^{-1}}{1-2[\frac{1}{4}\cos(\pi/2)]z^{-1}+\left(\frac{1}{4}\right)^2 z^{-2}};\ |z| > \frac{1}{4} \\
&= -z\frac{d}{dz}\left(\frac{\frac{1}{4}z^{-1}}{1+\frac{1}{16}z^{-2}}\right) - \frac{12\,z^{-1}}{1+\frac{1}{16}z^{-2}} = \frac{4z^{-1}-\frac{1}{4}z^{-3}}{1+\frac{1}{8}z^{-2}+\frac{1}{256}z^{-4}} - \frac{12\,z^{-1}}{1+\frac{1}{16}z^{-2}};\ |z| > \frac{1}{4} \\
&= \frac{-8z^{-1}-\frac{3}{2}z^{-3}-\frac{1}{16}z^{-5}}{1+\frac{3}{16}z^{-2}+\frac{3}{256}z^{-4}+\frac{1}{4096}z^{-6}};\ |z| > \frac{1}{4}
\end{aligned}
$$

Matlab script:

```
%% P0403e.m
clc; close all; b = [0 -8 0 -1.5 0 -1/16]; a = [1 0 3/16
0 3/256 0 1/(256*16)];
[delta,n1] = impseq(0,0,9); xb1 = filter(b,a,delta);
[u,n2] = stepseq(0,0,9);xb2 = (((n2-3).*((1/4).^(n2-
2))).*cos((pi/2)*(n2-1))).*u;
error = max(abs(xb1-xb2))
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0403e');
[Hz,Hp,Hl] = zplane(b,a); set(Hz,'linewidth',1);
set(Hp,'linewidth',1);
title('Pole-Zero plot','FontSize',12);
print -deps2 ../epsfiles/P0403e;
error =
   2.9392e-15
```

The pole-zero plot is shown in Figure 4.5.

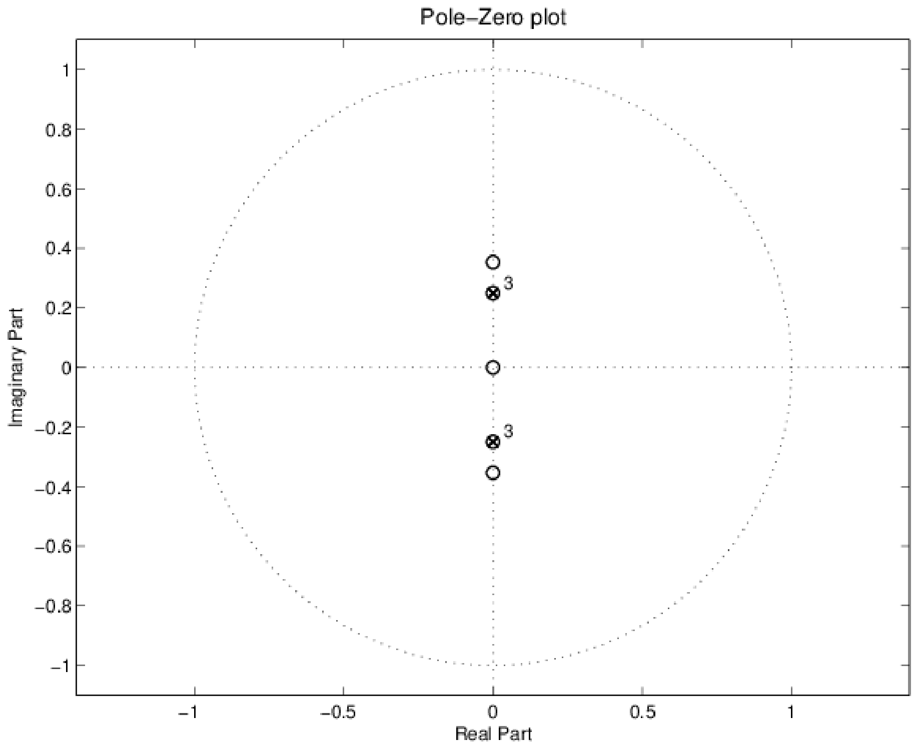


Figure 4.5: Problem P4.3.5 pole-zero plot

P4.4

Let $x(n)$ be a complex-valued sequence with the real part $x_R(n)$ and the imaginary part $x_I(n)$.
1. Prove the following z-transform relations:

$$X_R(z) \triangleq \mathcal{Z}[x_R(n)] = \frac{X(z) + X^*(z^*)}{2} \quad \text{and} \quad X_I(z) \triangleq \mathcal{Z}[x_I(n)] = \frac{X(z) - X^*(z^*)}{2}$$

2. Verify these relations for x(n) = exp {(-1 + j0.2π)n} u(n).

Solutions

1. The z-transform relations for real and imaginary parts.: Consider

$$X_R(z) \triangleq \mathcal{Z}[x_R(n)] = \mathcal{Z}\left[\frac{x(n) + x^*(n)}{2}\right] = \frac{\mathcal{Z}[x(n)] + \mathcal{Z}[x^*(n)]}{2}$$
$$= \frac{X(z) + X^*(z^*)}{2} \tag{4.1}$$

and

$$X_I(z) \triangleq \mathcal{Z}[x_I(n)] = \mathcal{Z}\left[\frac{x(n) - x^*(n)}{2}\right] = \frac{\mathcal{Z}[x(n)] - \mathcal{Z}[x^*(n)]}{2}$$
$$= \frac{X(z) - X^*(z^*)}{2} \tag{4.2}$$

2. Verification using $x(n) = \exp\{(-1 + j0.2\pi)n\}\, u(n)$: Consider

$$x(n) = \exp\{(-1 + j0.2\pi)n\}\, u(n) = e^{-n}\, e^{j0.2\pi n} u(n)$$
$$= e^{-n}\, \{\cos(0.2\pi n)u(n) + j\, \sin(0.2\pi n)u(n)\}$$

Hence the real and imaginary parts of $x(n)$, respectively, are

$$x_R(n) = e^{-n} \cos(0.2\pi n)u(n) = (1/e)^n \cos(0.2\pi n)u(n) \quad (4.3)$$
$$x_I(n) = e^{-n} \sin(0.2\pi n)u(n) = (1/e)^n \sin(0.2\pi n)u(n) \quad (4.4)$$

with z-transforms , respectively,

$$X_R(z) = \frac{1 - [(1/e)\cos(0.2\pi)]z^{-1}}{1 - [(2/e)\cos(0.2\pi)]z^{-1} + (1/e^2)z^{-2}} = \frac{1 - 0.2976z^{-1}}{1 - 0.5952z^{-1} + 0.1353z^{-2}}, \; |z| > 1/e \quad (4.5)$$
$$X_I(z) = \frac{[(1/e)\sin(0.2\pi)]z^{-1}}{1 - [(2/e)\cos(0.2\pi)]z^{-1} + (1/e^2)z^{-2}} = \frac{0.2162z^{-1}}{1 - 0.5952z^{-1} + 0.1353z^{-2}}, \; |z| > 1/e \quad (4.6)$$

The z-transform of $x(n)$ is

$$X(z) = \mathcal{Z}\left[\left(e^{-1+j0.2\pi}\right)^n u(n)\right] = \frac{1}{1 - e^{-1+j0.2\pi} z^{-1}}, \; |z| > 1/e \quad (4.7)$$

Substituting (4.7) in (4.1),

$$\begin{aligned} X_R(z) &= \frac{1}{2}\left[\frac{1}{1 - e^{-1+j2\pi} z^{-1}} + \left(\frac{1}{1 - e^{-1+j2\pi} z^{-*}}\right)^*\right] \\ &= \frac{1}{2}\left[\frac{1}{1 - e^{-1+j2\pi} z^{-1}} + \frac{1}{1 - e^{-1-j2\pi} z^{-1}}\right] = \frac{1}{2}\left[\frac{2 - 0.5952z^{-1}}{1 - 0.5952z^{-1} + 0.1353z^{-2}}\right] \\ &= \frac{1 - 0.2976z^{-1}}{1 - 0.5952z^{-1} + 0.1353z^{-2}}, \; |z| > e^{-1} \end{aligned} \quad (4.8)$$

as expected in (4.5). Similarly, Substituting (4.7) in (4.2),

$$\begin{aligned} X_I(z) &= \frac{1}{2}\left[\frac{1}{1 - e^{-1+j2\pi} z^{-1}} - \left(\frac{1}{1 - e^{-1+j2\pi} z^{-*}}\right)^*\right] \\ &= \frac{1}{2}\left[\frac{1}{1 - e^{-1+j2\pi} z^{-1}} - \frac{1}{1 - e^{-1-j2\pi} z^{-1}}\right] = \frac{1}{2}\left[\frac{0.2162z^{-1}}{1 - 0.5952z^{-1} + 0.1353z^{-2}}\right] \\ &= \frac{0.2162z^{-1}}{1 - 0.5952z^{-1} + 0.1353z^{-2}}, \; |z| > e^{-1} \end{aligned} \quad (4.9)$$

as expected in (4.6).

P4.5

The z-transform of $x(n)$ is $X(z) = 1/(1 + 0.5z^{-1})$, $|z| \geq 0.5$. Determine the z-transforms of the following sequences and indicate their region of convergence.

1. $x_1(n) = x(3 - n) + x(n - 3)$
2. $x_2(n) = (1 + n + n^2)x(n)$
3. $x_3(n) = (\frac{1}{2})^n x(n - 2)$
4. $x_4(n) = x(n + 2) * x(n - 2)$
5. $x_5(n) = \cos(\pi n/2)x^*(n)$

Solutions

1. The z-transforms of $x_1(n) = x(3 - n) + x(n - 3)$:

$$
\begin{aligned}
X_1(z) &= \mathcal{Z}[x_1(n)] = \mathcal{Z}[x(3-n) + x(n-3)] = \mathcal{Z}[x\{-(n-3)\}] + \mathcal{Z}[x(n-3)] \\
&= z^{-3}X(1/z) + z^{-3}X(z) = z^{-3}\left[\frac{1}{1+0.5z} + \frac{1}{1+0.5z^{-1}}\right], \ 0.5 < |z| < 2 \\
&= \frac{0.5z^{-3} + 2z^{-4} + 0.5z^{-5}}{0.5 + 1.25z^{-1} + 0.5z^{-2}}, \ 0.5 < |z| < 2
\end{aligned}
$$

2. The z-transforms of $x_2(n) = (1 + n + n^2)x(n)$:

$$
\begin{aligned}
X_2(z) &= \mathcal{Z}\left[\left(1+n+n^2\right)x(n)\right] = \mathcal{Z}\left[x(n) + n\,x(n) + n^2x(n)\right] = X(z) - z\frac{d}{dz}X(z) + z^2\frac{d^2}{dz^2}X(z) \\
&= \frac{1}{1+0.5z^{-1}} - \frac{0.5z^{-1}}{\left(1+0.5z^{-1}\right)^2} - \frac{z^{-1}+0.5z^{-2}}{\left(1+0.5z^{-1}\right)^4} = \frac{1-0.25z^{-2}}{\left(1+0.5z^{-1}\right)^4}, \ |z| > 0.5
\end{aligned}
$$

3. The z-transforms of $x_3(n) = \left(\frac{1}{2}\right)^n x(n - 2)$:

$$
\begin{aligned}
X_3(z) &= \mathcal{Z}\left[\left(\frac{1}{2}\right)^n x(n-2)\right] = \mathcal{Z}[x(n-2)]\big|_{\left(\frac{1}{2}\right)^{-1}z} = \mathcal{Z}[x(n-2)]\big|_{2z} \\
&= \left[z^{-2}X(z)\right]\big|_{2z} = \left[\frac{z^{-2}}{1+0.5z^{-1}}, \ |z| > 0.5\right]\Bigg|_{2z} = \frac{0.25z^{-2}}{1+0.25z^{-1}}, \ |z| > 0.25
\end{aligned}
$$

4. The z-transforms of $x_4(n) = x(n + 2) * x(n - 2)$:

$$
\begin{aligned}
X_4(z) &= \mathcal{Z}[x(n+2) * x(n-2)] = \left\{z^2X(z)\right\}\left\{z^{-2}X(z)\right\} = X^2(z) \\
&= \frac{1}{\left(1+0.5z^{-1}\right)^2}, \ |z| > 0.5
\end{aligned}
$$

5. The z-transforms of $x_5(n) = \cos(\pi n/2)x^*(n)$:

$$
\begin{aligned}
X_5(z) &= \mathcal{Z}\left[\cos(\pi n/2)x^*(n)\right] = \mathcal{Z}\left[\left(\frac{e^{j\pi n/2} + e^{-j\pi n/2}}{2}\right)x^*(n)\right] \\
&= \frac{1}{2}\left(\mathcal{Z}\left[e^{j\pi n/2}x^*(n)\right] + \mathcal{Z}\left[e^{-j\pi n/2}x^*(n)\right]\right) \\
&= \frac{1}{2}\left(\mathcal{Z}\left[x^*(n)\right]\big|_{e^{-j\pi/2}z} + \mathcal{Z}\left[x^*(n)\right]\big|_{e^{j\pi/2}z}\right) \\
&= \frac{1}{2}\left[X^*\left(e^{j\pi/2}z^*\right) + X^*\left(e^{-j\pi/2}z^*\right)\right] \\
&= \frac{1}{2}\left[\frac{1}{1+0.5e^{-j\pi/2}z^{-1}} + \frac{1}{1+0.5e^{j\pi/2}z^{-1}}\right] \\
&= \frac{1}{1+0.25z^{-2}}, \ |z| > 0.5
\end{aligned}
$$

P4.6

Repeat Problem P4.5 if

$$X(z) = \frac{1 + z^{-1}}{1 + \frac{5}{6}z^{-1} + \frac{1}{6}z^{-2}};\ |z| > \frac{1}{2}$$

Solutions

1. The z-transforms of $x_1(n) = x(3 - n) + x(n - 3)$:

$$\begin{aligned} X_1(z) &= \mathcal{Z}[x_1(n)] = \mathcal{Z}[x(3-n) + x(n-3)] = \mathcal{Z}[x\{-(n-3)\}] + \mathcal{Z}[x(n-3)] \\ &= z^{-3}X(1/z) + z^{-3}X(z) = z^{-3}\left[\frac{1+z}{1+\frac{5}{6}z+\frac{1}{6}z^2} + \frac{1+z^{-1}}{1+\frac{5}{6}z^{-1}+\frac{1}{6}z^{-2}}\right],\ 0.5 < |z| < 2 \\ &= \frac{36z^{-1} + 72z^{-2} + 36z^{-3}}{6 + 35z^{-1} + 62z^{-2} + 35z^{-3} + 6z^{-4}},\ 0.5 < |z| < 2 \end{aligned}$$

2. The z-transforms of $x_2(n) = (1 + n + n^2)x(n)$:

$$X_2(z) = \mathcal{Z}\left[\left(1+n+n^2\right)x(n)\right] = \mathcal{Z}\left[x(n) + n\,x(n) + n^2x(n)\right] = X(z) - z\frac{d}{dz}X(z) + z^2\frac{d^2}{dz^2}X(z)$$

or $X_2(z) =$

$$\frac{1+\frac{17}{3}z^{-1}+\frac{1}{2}z^{-2}+\frac{1429}{108}z^{-3}+\frac{2399}{286}z^{-4}+\frac{215}{72}z^{-5}+\frac{829}{1944}z^{-6}-\frac{167}{1944}z^{-7}-\frac{7}{144}z^{-8}-\frac{2}{243}z^{-9}-\frac{1}{1944}z^{-10}}{1+5z^{-1}+\frac{137}{12}z^{-2}+\frac{425}{27}z^{-3}+\frac{6305}{432}z^{-4}+\frac{2694}{281}z^{-5}+\frac{1711}{374}z^{-6}+\frac{449}{281}z^{-7}+\frac{1258}{3103}z^{-8}+\frac{425}{5832}z^{-9}+\frac{137}{15552}z^{-10}+\frac{5}{7776}z^{-11}+\frac{1}{46656}z^{-12}},\ |z| > 0.5$$

3. The z-transforms of $x_3(n) = \left(\frac{1}{2}\right)^n x(n - 2)$:

$$\begin{aligned} X_3(z) &= \mathcal{Z}\left[\left(\frac{1}{2}\right)^n x(n-2)\right] = \mathcal{Z}[x(n-2)]|_{\left(\frac{1}{2}\right)^{-1}z} = \mathcal{Z}[x(n-2)]|_{2z} \\ &= \left[z^{-2}X(z)\right]\Big|_{2z} = [,\ |z| > 0.5]|_{2z} = \frac{\frac{1}{4}z^{-2} + \frac{1}{8}z^{-3}}{1 + \frac{5}{12}z^{-1} + \frac{1}{24}z^{-2}},\ |z| > 0.25 \end{aligned}$$

4. The z-transforms of $x_4(n) = x(n + 2) * x(n - 2)$:

$$\begin{aligned} X_4(z) &= \mathcal{Z}[x(n+2) * x(n-2)] = \left\{z^2X(z)\right\}\left\{z^{-2}X(z)\right\} = X^2(z) \\ &= \frac{1 + 2z^{-1} + z^{-2}}{1 + \frac{5}{3}z^{-1} + \frac{37}{36}z^{-2} + \frac{5}{18}z^{-3} + \frac{1}{36}z^{-4}},\ |z| > 0.5 \end{aligned}$$

5. The z-transforms of $x_5(n) = \cos(\pi n/2)x^*(n)$:

$$
\begin{aligned}
X_5(z) &= \mathcal{Z}\left[\cos(\pi n/2)x^*(n)\right] = \mathcal{Z}\left[\left(\frac{e^{j\pi n/2}+e^{-j\pi n/2}}{2}\right)x^*(n)\right] \\
&= \frac{1}{2}\left(\mathcal{Z}\left[e^{j\pi n/2}x^*(n)\right] + \mathcal{Z}\left[e^{-j\pi n/2}x^*(n)\right]\right) \\
&= \frac{1}{2}\left(\mathcal{Z}\left[x^*(n)\right]\big|_{e^{-j\pi/2}z} + \mathcal{Z}\left[x^*(n)\right]\big|_{e^{j\pi/2}z}\right) \\
&= \frac{1}{2}\left[X^*\left(e^{j\pi/2}z^*\right) + X^*\left(e^{-j\pi/2}z^*\right)\right] \\
&= \frac{1}{2}\left[\frac{1-jz^{-1}}{1-j\frac{5}{6}z^{-1}-\frac{1}{6}z^{-2}}\frac{1+jz^{-1}}{1+j\frac{5}{6}z^{-1}-\frac{1}{6}z^{-2}}\right] \\
&= \frac{2+\frac{4}{3}z^{-2}}{1+\frac{13}{36}z^{-2}+\frac{1}{36}z^{-4}},\ |z|>0.5
\end{aligned}
$$

P4.7

The inverse z-transform of $X(z)$ is $x(n) = (1/2)^n u(n)$. Using the z-transform properties, determine the sequences in each of the following cases.

1. $X_1(z) = \frac{z-1}{z}X(z)$
2. $X_2(z) = zX(z^{-1})$
3. $X_3(z) = 2X(3z) + 3X(z/3)$
4. $X_4(z) = X(z)X(z^{-1})$
5. $X_5(z) = z^2\frac{dX(z)}{dz}$

Solutions

1. $X_1(z) = \dfrac{z-1}{z}X(z)$: Consider

$$
\begin{aligned}
x_1(n) &= \mathcal{Z}^{-1}\left[X_1(z)\right] = \mathcal{Z}^{-1}\left[\left(1-\frac{1}{z}\right)X(z)\right] = \mathcal{Z}^{-1}\left[X(z) - z^{-1}X(z)\right] \\
&= x(n) - x(n-1) = 0.5^n u(n) - 0.5^{n-1}u(n-1) = 1 - 0.5^n u(n-1)
\end{aligned}
$$

2. $X_2(z) = zX(z^{-1})$: Consider

$$
\begin{aligned}
x_2(n) &= \mathcal{Z}^{-1}\left[X_2(z)\right] = \mathcal{Z}^{-1}\left[zX(z^{-1})\right] = \mathcal{Z}^{-1}\left[X(z^{-1})\right]\big|_{n\to(n+1)} = \mathcal{Z}^{-1}\left[X(z)\right]\big|_{n\to-(n+1)} \\
&= (0.5)^{-(n+1)}u(-n-1) = 2^{n+1}u(-n-1)
\end{aligned}
$$

3. $X_3(z) = 2X(3z) + 3X(z/3)$: Consider

$$x_3(n) = \mathcal{Z}^{-1}[X_3(z)] = \mathcal{Z}^{-1}[2X(3z) + 3X(z/3)] = 2(3^{-n})x(n) + 3(3^n)x(n)$$
$$= 2(3^{-n})(2^{-n})u(n) + 3(3^n)(2^{-n})u(n) = \left[2\left(\frac{1}{6}\right)^n + 3\left(\frac{3}{2}\right)^n\right]u(n)$$

4. $X_4(z) = X(z)X(z^{-1})$: Consider

$$x_4(n) = \mathcal{Z}^{-1}[X_4(z)] = \mathcal{Z}^{-1}\left[X(z)X(z^{-1})\right] = x(n) * x(-n)$$
$$= \left[0.5^n u(n)\right] * \left[2^n u(-n)\right] = \sum_{k=-\infty}^{\infty} (0.5)^k u(k) 2^{n-k} u(-n+k)$$
$$= \begin{cases} 2^n \sum_{k=0}^{\infty} (0.5)^k 2^{-k}, & n < 0; \\ 2^n \sum_{k=n}^{\infty} (0.5)^k 2^{-k}, & n \ge 0. \end{cases} = \begin{cases} 2^n \sum_{k=-\infty}^{\infty} (0.25)^k, & n < 0; \\ 2^n 2^{-2n} \sum_{k=-\infty}^{\infty} (0.25)^k, & n \ge 0. \end{cases}$$
$$= \frac{4}{3} 2^{|n|}$$

5. $X_5(z) = z^2 \dfrac{dX(z)}{dz}$: Consider

$$x_5(n) = \mathcal{Z}^{-1}[X_5(z)] = \mathcal{Z}^{-1}\left[z^2 \frac{dX(z)}{dz}\right] = n^2 x(n)$$
$$= n^2 (1/2)^n u(n)$$

P4.8

If sequences $x_1(n)$, $x_2(n)$, and $x_3(n)$ are related by $x_3(n) = x_1(n) * x_2(n)$, then

$$\sum_{n=-\infty}^{\infty} x_3(n) = \left(\sum_{n=-\infty}^{\infty} x_1(n)\right)\left(\sum_{n=-\infty}^{\infty} x_2(n)\right)$$

1. Prove this result by substituting the definition of convolution in the left-hand side.
2. Prove this result using the convolution property.
3. Verify this result using MATLABand choosing any two random sequences $x_1(n)$, and $x_2(n)$.

Solutions

1. Proof using the definition of convolution:

$$\sum_{n=-\infty}^{\infty} x_3(n) = \left(\sum_{n=-\infty}^{\infty} x_1(n)\right)\left(\sum_{n=-\infty}^{\infty} x_2(n)\right)$$

as expected.

2. Proof using the convolution property:

$$\sum_{n=-\infty}^{\infty} x_3(n) = \sum_{n=-\infty}^{\infty} x_1(n) * x_2(n) = \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} x_1(k)x_2(n-k) = \left(\sum_{k=-\infty}^{\infty} x_1(k)\right) \sum_{n=-\infty}^{\infty} x_2(n-k)$$
$$= \left(\sum_{n=-\infty}^{\infty} x_1(n)\right)\left(\sum_{n=-\infty}^{\infty} x_2(n)\right)$$

as expected.

3. Matlab verification:

```
% P4.8
% P0408.m
clc; close all;
N = 1000; n1 = [0:N]; x1 = rand(1,length(n1));
n2 = [0:N]; x2 = rand(1,length(n2)); [x3,n3] =
conv_m(x1,n1,x2,n2);
sumx1 = sum(x1); sumx2 = sum(x2); sumx3 = sum(x3);
error = max(abs(sumx3-sumx1*sumx2))

error =
   5.8208e-11
```

P4.9

Determine the results of the following polynomial operations using MATLAB.

1. $X_1(z) = (1 - 2z^{-1} + 3z^{-2} - 4z^{-3})(4 + 3z^{-1} - 2z^{-2} + z^{-3})$
2. $X_2(z) = (z^2 - 2z + 3 + 2z^{-1} + z^{-2})(z^3 - z^{-3})$
3. $X_3(z) = (1 + z^{-1} + z^{-2})^3$
4. $X_4(z) = X_1(z)X_2(z) + X_3(z)$
5. $X_5(z) = (z^{-1} - 3z^{-3} + 2z^{-5} + 5z^{-7} - z^{-9})(z + 3z^2 + 2z^3 + 4z^4)$

Solutions

1. $X_1(z) = (1 - 2z^{-1} + 3z^{-2} - 4z^{-3})(4 + 3z^{-1} - 2z^{-2} + z^{-3})$

```
% P4.9
%% P0409a.m
clc; close all;
n1 = [0:3]; y1 = [1 -2 3 -4]; n2 = [0:3]; y2 = [4 3 -2
1];
[x1,n] = conv_m(y1,n1,y2,n2)

x1 =
     4    -5     4    -2   -20    11    -4
n =
     0     1     2     3     4     5     6
```

Hence

$$X_1(z) = 4 - 5z^{-1} + 4z^{-2} - 2z^{-3} - 20z^{-4}11 + z^{-5} - 4z^{-6}$$

2. $X_2(z) = (z^2 - 2z + 3 + 2z^{-1} + z^{-2})(z^3 - z^{-3})$

```
%% P0409b.m
clc; close all;
n1 = [-2:2]; y1 = [1 -2 3 2 1]; n2 = [-3:3]; y2 = [1 0 0
0 0 0 1];
[x2,n] = conv_m(y1,n1,y2,n2)
x2 =
     1    -2     3     2     1     0     1    -2     3     2     1
n =
    -5    -4    -3    -2    -1     0     1     2     3     4     5
```

Hence

$$X_2(z) = z^5 - 2z^4 + 3z^3 + 2z^4 + z + z^{-1} - 2z^{-2} + 3z^{-3} + 2z^{-4} + z^{-5}$$

3. $X_3(z) = (1 + z^{-1} + z^{-2})^3$

```
%% P0409c.m
clc; close all;
n1 = [0 1 2]; y1 = [1 1 1]; [y2,n2] =
conv_m(y1,n1,y1,n1);
[x3,n] = conv_m(y1,n1,y2,n2)
x3 =
     1     3     6     7     6     3     1
n =
     0     1     2     3     4     5     6
```

Hence

$$X_3(z) = 1 + 3z^{-1} + 6z^{-2} + 7z^{-3} + 6z^{-4} + 3z^{-5} + z^{-6}$$

4. $X_4(z) = X_1(z)X_2(z) + X_3(z)$

```
%% P0409d.m
clc; close all;
n11 = [0:3]; y11 = [1 -2 3 -4]; n12 = [0:3]; y12 = [4 3 -
2 1];
[y13,n13] = conv_m(y11,n11,y12,n12);
n21 = [-2:2]; y21 = [1 -2 3 2 1]; n22 = [-3:3]; y22 = [1
0 0 0 0 0 1];
[y23,n23] = conv_m(y21,n21,y22,n22);
n31 = [0 1 2]; y31 = [1 1 1];
[y32,n32] = conv_m(y31,n31,y31,n31);
```

```
[y33,n33] = conv_m(y31,n31,y32,n32);
[y41,n41] = conv_m(y13,n13,y23,n23);
[x4,n] = sigadd(y41,n41,y33,n33)

x4 =
  Columns 1 through 14
     4   -13    26   -17   -10    49   -79    -8    23    -8   -
11    49   -86    -1
  Columns 15 through 17
   -10     3    -4
n =
  Columns 1 through 14
    -5    -4    -3    -2    -1     0     1     2     3     4     5
6     7     8
  Columns 15 through 17
     9    10    11
```

Hence

$$X_4(z) = 4z^5 - 13z^4 + 26z^3 - 17z^2 - 10z^1 + 49 - 79z^{-1} - 8z^{-2} + 23z^{-3} - 8z^{-4} - 11z^{-5} + 49z^{-6} - 86z^{-7} - z^{-8} - 10z^{-9} + 3z^{-10} - 4z^{-11}$$

5. $X_5(z) = (z^{-1} - 3z^{-3} + 2z^{-5} + 5z^{-7} - z^{-9})(z + 3z^2 + 2z^3 + 4z^4)$

```
%% P0409e.m
clc; close all;
n1 = [0:9]; y1 = [0 1 0 -3 0 2 0 5 0 -1]; n2 = [-4:0]; y2
= [4 2 3 1 0];
[x5,n] = conv_m(y1,n1,y2,n2)
x5 =
     0     4     2    -9    -5    -1     1    26    12    11
3    -3    -1     0
n =
    -4    -3    -2    -1     0     1     2     3     4     5     6
7     8     9
```

Hence

$$X_5(z) = 4z^3 + 2z^2 - 9z^1 - 5 - z^{-1} + z^{-2} + 26z^{-3} + 12z^{-4} + 11z^{-5} + 3z^{-6} - 3z^{-7} - z^{-8}$$

P4.10

The **deconv** function is useful in dividing two causal sequences. Write a MATLAB function **deconv_m** to divide two noncausal sequences (similar to the **conv** function). The format of this function should be

function [p,np,r,nr] = deconv_m(b,nb,a,na)

% Modified deconvolution routine for noncausal sequences

% function [p,np,r,nr] = deconv_m(b,nb,a,na)

% %

p = polynomial part of support np1 <= n <= np2

% np = [np1, np2]

% r = remainder part of support nr1 <= n <= nr2

% nr = [nr1, nr2]

% b = numerator polynomial of support nb1 <= n <= nb2

% nb = [nb1, nb2]

% a = denominator polynomial of support na1 <= n <= na2

% na = [na1, na2]

%

Check your function on the following operartion

$$\frac{z^2+z+1+z^{-1}+z^{-2}+z^{-3}}{z+2+z^{-1}} = (z-1+2z^{-1}-2z^{-2}) + \frac{3z^{-2}+3z^{-3}}{z+2+z^{-1}}$$

Solutions

The Matlab function **deconv_m**:

```
function [p,np,r,nr] = deconv_m(b,nb,a,na)
% Modified deconvolution routine for noncausal sequences
% function [p,np,r,nr] = deconv_m(b,nb,a,na)
%
% p = polynomial part of support np1 <= n <= np2
% np = [np1, np2]
% r = remainder part of support nr1 <= n <= nr2
% nr = [nr1, nr2]
% b = numerator polynomial of support nb1 <= n <= nb2
% nb = [nb1, nb2]
% a = denominator polynomial of support na1 <= n <= na2
% na = [na1, na2]
%
[p,r] = deconv(b,a);
np1 = nb(1) - na(1); np2 = np1 + length(p)-1; np =
[np1:np2];
nr1 = nb(1); nr2 = nr1 + length(r)-1; nr = [nr1:nr2];
```

Matlab verification:

```
% P4.10
% P0410.m
```

```
clc; close all;
nb = [-2:3]; b = [1 1 1 1 1 1]; na = [-1:1]; a = [1 2 1];
[p,np,r,nr] = deconv_m(b,nb,a,na)

p =
     1    -1     2    -2
np =
    -1     0     1     2
r =
     0     0     0     0     3     3
nr =
    -2    -1     0     1     2     3
```

Hence

$$\frac{z^2+z+1+z^{-1}+z^{-2}+z^{-3}}{z+2+z^{-1}} = (z-1+2z^{-1}-2z^{-2}) + \frac{3z^{-2}+3z^{-3}}{z+2+z^{-1}}$$

P4.11

Determine the following inverse z-transforms using the partial fraction expansion method.

1. $X_1(z) = (1-z^{-1}-4z^{-2}+4z^{-3})/(1-\frac{11}{4}z^{-1}+\frac{13}{8}z^{-2}-\frac{1}{4}z^{-3})$. The sequence is rightsided.
2. $X_2(z) = (1+z^{-1}-4z^{-2}+4z^{-3})/(1-\frac{11}{4}z^{-1}+\frac{13}{8}z^{-2}-\frac{1}{4}z^{-3})$. The sequence is absolutely summable.
3. $X_3(z) = (z^3-3z^2+4z+1)/(z^3-4z^2+z-0.16)$. The sequence is leftsided.
4. $X_4(z) = z/(z^3+2z^2+1.25z+0.25)$, $|z|>1$
5. $X_5(z) = z/(z^2-0.25)^2$, $|z|<0.5$

Solutions

1. $X_1(z) = (1-z^{-1}-4z^{-2}+4z^{-3})/(1-\frac{11}{4}z^{-1}+\frac{13}{8}z^{-2}-\frac{1}{4}z^{-3})$. The sequence is right-sided. MATLAB script:

```
% P4.11
%% P0411a: Inverse z-Transform of X1(z)
clc; close all;
b1 = [1,-1,-4,4]; a1 = [1,-11/4,13/8,-1/4];
[R,p,k] = residuez(b1,a1)

R =
    0.0000
  -10.0000
   27.0000
p =
```

```
    2.0000
    0.5000
    0.2500
k =
   -16
```

or

$$X_1(z) = \frac{1 - z^{-1} - 4z^{-2} + 4z^{-3}}{1 - \frac{11}{4}z^{-1} + \frac{13}{8}z^{-2} - \frac{1}{4}z^{-3}} = -16 + \frac{0}{1 - 2z^{-1}} - \frac{10}{1 - 0.5z^{-1}} + \frac{27}{1 - 0.25z^{-1}}, \ |z| > 0.5$$

Note that from the second term on the right, there is a pole-zero cancellation. Hence

$$x_1(n) = -16\delta(n) - 10(0.5)^n u(n) + 27(0.25)^n u(n)$$

2. $X_2(z) = (1 + z^{-1} - 4z^{-2} + 4z^{-3})/(1 - \frac{11}{4}z^{-1} + \frac{13}{8}z^{-2} - \frac{1}{4}z^{-3})$. The sequence is absolutely summable. MATLAB script:

```
%% P0411b: Inverse z-Transform of X2(z)
clc; close all;
b2 = [1,1,-4,4]; a2 = [1,-11/4,13/8,-1/4];
[R,p,k] = residuez(b2,a2)

R =
    1.5238
  -12.6667
   28.1429
p =
    2.0000
    0.5000
    0.2500
k =
   -16
```

or

$$X_2(z) = \frac{1 - z^{-1} - 4z^{-2} + 4z^{-3}}{1 - \frac{11}{4}z^{-1} + \frac{13}{8}z^{-2} - \frac{1}{4}z^{-3}} = -16 + \frac{1.5238}{1 - 2z^{-1}} - \frac{12.6667}{1 - 0.5z^{-1}} + \frac{28.1429}{1 - 0.25z^{-1}}, \ 0.5 < |z| > 2$$

Hence

$$x_2(n) = -16\delta(n) - 1.5238(2)^n u(-n-1) - 12.6667(0.5)^n u(n) + 28.1429(0.25)^n u(n)$$

3. $X_3(z) = (z^3 - 3z^2 + 4z + 1)/(z^3 - 4z^2 + z - 0.16)$. The sequence is left-sided. Consider

Matlab script for the PFE:

```
%% P0411c: Inverse z-Transform of X3(z)
clc; close all;
```

```
b3 = [1,-3,4,1]; a3 = [1,-4,1,-0.16];
[R,p,k] = residuez(b3,a3)

r = abs(p(2))
[b,a] = residuez(R(2:3),p(2:3),[])

R =
   0.5383 + 0.0000i
   3.3559 + 5.7659i
   3.3559 - 5.7659i
p =
   3.7443 + 0.0000i
   0.1278 + 0.1625i
   0.1278 - 0.1625i
k =
   -6.2500
r =
    0.2067
b =
    6.7117   -2.7313
a =
    1.0000   -0.2557    0.0427
```

or

$$X_3(z) = -6.25 + \frac{0.5383}{1 - 3.7443z^{-1}} + \frac{3.3559 + j5.7659}{1 - (0.1278 + j0.1625)z^{-1}} + \frac{3.3559 - j5.7659}{1 - (0.1278 - j0.1625)z^{-1}}, \; |z| < 0.2067$$

Hence

$$\begin{aligned} x_3(n) = &-6.25\delta(n) - 0.5383(3.7443)^n u(-n-1) \\ &- (3.3559 + j5.7659)(0.1278 + j0.1625)^n u(-n-1) \\ &- (3.3559 - j5.7659)(0.1278 - j0.1625)^n u(-n-1) \end{aligned}$$

4. $X_4(z) = z/(z^3 + 2z^2 + 1.25z + 0.25)$, $|z| > 1$. Consider

$$X_4(z) = \frac{z}{z^3 + 2z^2 + 1.25z + 0.25} = \frac{z^{-2}}{1 + 2z^{-1} + 1.25z^{-2} + 0.25z^{-3}}$$

Matlab script for the PFE:

```
%% P0411d: Inverse z-Transform of X4(z)
clc; close all;
b4 = [0,0,1]; a4 = [1,2,1.25,0.25];
[R,p,k] = residuez(b4,a4)
```

```
R =
   4.0000 + 0.0000i
   0.0000 + 0.0000i
  -4.0000 + 0.0000i
p =
  -1.0000 + 0.0000i
  -0.5000 + 0.0000i
  -0.5000 - 0.0000i
k =
     []
```

or

$$X_4(z) = \frac{4}{1+z^{-1}} - \frac{4}{\left(1+0.5z^{-1}\right)^2} = \frac{4}{1+z^{-1}} - 8z\frac{0.5z^{-1}}{\left(1+0.5z^{-1}\right)^2}, \ |z| > 1$$

Hence

$$x_4(n) = 4(-1)^n u(n) - 8(n+1)(0.5)^{n+1} u(n+1)$$

5. $X_5(z) = z/(z^2 - 0.25)^2$, $|z| < 0.5$. Consider

$$X_5(z) = \frac{z}{(z^2 - 0.25)^2} = \frac{z^{-3}}{(1 - 0.25z^{-2})^2}, \ |z| < 0.5$$

Matlab script for PFE:

```
%% P0411e: Inverse z-Transform of X5(z)
clc; close all;
b5 = [0,0,0,1]; a5 = conv([1,0,-0.25],[1,0,-0.25]);
[R,p,k] = residuez(b5,a5)
```

```
R =
   4.0000 - 0.0000i
  -2.0000 + 0.0000i
  -4.0000 - 0.0000i
   2.0000 + 0.0000i
p =
  -0.5000 + 0.0000i
  -0.5000 + 0.0000i
   0.5000 + 0.0000i
   0.5000 - 0.0000i
k =
     []
```

or

$$X_5(z) = \frac{4}{1+0.5z^{-1}} - 2\frac{1}{(1+0.5z^{-1})^2} - 4\frac{1}{1-0.5z^{-1}} + 2\frac{1}{(1-0.5z^{-1})^2}, \; |z| < 0.5$$
$$= \frac{4}{1-(-0.5)z^{-1}} + 4z\frac{(-0.5)z^{-1}}{[1-(-0.5)z^{-1}]^2} - 4\frac{1}{1-0.5z^{-1}} + 4z\frac{0.5z^{-1}}{(1-0.5z^{-1})^2}, \; |z| < 0.5$$

Hence

$$\begin{aligned} x_5(n) &= -4(-0.5)^n u(-n-1) - 4(n+1)(-0.5)^n u[-(n+1)-1] + 4(0.5)^n u(-n-1) \\ &\quad - 4(n+1)(0.5)^n u[-(n+1)-1] \\ &= -4(-0.5)^n u(-n-1) - 4(n+1)(-0.5)^n u[-n-2] + 4(0.5)^n u(-n-1) \\ &\quad - 4(n+1)(0.5)^n u[-n-2] \\ &= 4(0.5)^n \left[1-(-1)^n\right] u(-n-1) - 4(n+1)(0.5)^n \left[1+(-1)^n\right] u(-n-2) \end{aligned}$$

P4.12

Consider the sequence

$$x(n) = A_c(r)^n \cos(\pi v_0 n)u(n) + A_s(r)^n \sin(\pi v_0 n)u(n) \qquad \textbf{(4.30)}$$

The z-transform of this sequence is a 2-order (proper) rational function that contains a complex-conjugate pole pair. The objective of this problem is to develop a MATLAB function that can be used to obtain the inverse z-transform of such a rational function so that the inverse does not contain any complex numbers.

1. Show that the z-transform of $x(n)$ in (4.30) is given by

$$X(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}; \quad |z| > |r| \qquad \textbf{(4.31)}$$

Where

$$b_0 = A_c;\; b_1 = r[A_s \sin(\pi v_0) - A_c \cos(\pi v_0)];\; a_1 = -2r\cos(\pi v_0);\; a_2 = r^2 \qquad \textbf{(4.32)}$$

2. Using (4.32), determine the signal parameters A_c, A_s, r, and v_0 in terms of the rational function parameters b_0, b_1, a_1, and a_2.

3. Using your results in part b above, design a MATLABfunction, inv_CC_PP, that computes signal parameters using the rational function parameters. The format of this function should be:

function [As,Ac,r,v0] = inv_CC_PP(b0,b1,a1,a2)

Solutions

Consider the sequence given below:

$$x(n) = A_c(r)^n \cos(\pi v_0 n)u(n) + A_s(r)^n \sin(\pi v_0 n)u(n) \qquad (4.10)$$

1. The z-transform of $x(n)$ in (4.10): Taking z-transform of (4.10),

$$X(z) = A_c \frac{1 - r\ \cos(\pi v_0)z^{-1}}{1 - 2r\ \cos(\pi v_0)z^{-1} + r^2 z^{-2}} + A_s \frac{r\ \sin(\pi v_0)z^{-1}}{1 - 2r\ \cos(\pi v_0)z^{-1} + r^2 z^{-2}}$$
$$= \frac{A_c + r\,[A_s \sin(\pi v_0) - A_c \cos(\pi v_0)]\,z^{-1}}{1 - 2r\ \cos(\pi v_0)z^{-1} + r^2 z^{-2}} \triangleq \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}}; \quad |z| > |r|$$

Comparing the last step above, we have

$$b_0 = A_c;\ b_1 = r[A_s \sin(\pi v_0) - A_c \cos(\pi v_0)];\ a_1 = -2r\cos(\pi v_0);\ a_2 = r^2 \qquad (4.11)$$

2. The signal parameters A_c, A_s, r, and v_0 in terms of the rational function parameters b_0, b_1, a_1, and a_2: Using (4.11 and solving parameters in the following order: A_c, then r, then v_0, and finally A_s, we obtain

$$A_c = b_0; \quad r = \sqrt{a_2}; \quad v_0 = \frac{\arccos(-a_1/2r)}{\pi}; \quad A_s = \frac{2b_1 - a_1 b_0}{\sqrt{4a_2 - a_1^2}}$$

3. Matlab function **inv_CC_PP**:

```
function [Ac,As,r,v0] = inv_CC_PP(b0,b1,a1,a2)
% [Ac,As,r,v0] = inv_CC_PP(b0,b1,a1,a2)
Ac = b0;
r = sqrt(a2);
w0 = acos(-a1/(2*r));
As = (b1/r+Ac*cos(w0))/sin(w0);
v0 = w0/(pi);
```

P4.13

Suppose $X(z)$ is given as follows:

$$X(z) = \frac{2 + 3z^{-1}}{1 - z^{-1} + 0.81z^{-2}}, \quad |z| > 0.9$$

1. Using the MATLAB function **inv_CC_PP** given in Problem P4.12, determine $x(n)$ in a form that contains no complex numbers.
2. Using MATLAB, compute the first 20 samples of $x(n)$, and compare them with your answer in the above part.

Solutions

1. The signal $x(n)$ in a form that contains no complex numbers: Matlab script:

```
% P4.13
%% P0413a.m
clc; close all;
b0 = 2; b1 = 3; a1 = -1; a2 = 0.81;
[Ac,As,r,v0] = inv_CC_PP(b0,b1,a1,a2);
disp(sprintf('\nx(n) = %1i*(%3.1f)^n*cos(%5.4f*pi*n)u(n)
',Ac,r,v0));
```

```
disp(sprintf('
+ %5.4f*(%3.1f)^n*sin(%5.4f*pi*n)u(n)\n',As,r,v0));
% x(n) = 2*(0.9)^n*cos(0.3125*pi*n)u(n)
% + 5.3452*(0.9)^n*sin(0.3125*pi*n)u(n);
```

2. Matlab verification:

```
%% P0413b.m
n = 0:20; x = Ac*(r.^n).*cos(v0*pi*n) +
As*(r.^n).*sin(v0*pi*n);
y = filter([b0,b1],[1,a1,a2],impseq(0,0,20));
error = abs(max(x-y))

x(n) = 2*(0.9)^n*cos(0.3125*pi*n)u(n)
 + 5.3452*(0.9)^n*sin(0.3125*pi*n)u(n)
error =
   1.7764e-15
```

P4.14

The z-transform of a causal sequence is given as

$$X(z) = \frac{-2 + 5.65z^{-1} - 2.88z^{-2}}{1 - 0.1z^{-1} + 0.09z^{-2} + 0.648z^{-3}} \qquad \textbf{(4.33)}$$

which contains a complex-conjugate pole pair as well as a real-valued pole.

1. Using the **residuez** function express (4.33) as

$$X(z) = \frac{(\quad) + (\quad)z^{-1}}{1 + (\quad)z^{-1} + (\quad)z^{-2}} + \frac{(\quad)}{1 + (\quad)z^{-1}} \qquad \textbf{(4.34)}$$

Note that you will have to use the **residuez** function in both directions.

2. Now using your function **inv_CC_PP** and the inverse of the real-valued pole factor, determine the causal sequence $x(n)$ from the $X(z)$ in (4.34) so that it contains no complex numbers.

Solutions

1. Rearrangement of $X(z)$ into a first- and second-order sections: Matlab Script:

```
% P4.14
%% P0414a.m
clc; close all;
b = [-2 5.65 -2.88]; a = [1 -0.1 .09 0.648]; [R,p,k] =
residuez(b,a)
[b1,a1] = residuez(R(1:2),p(1:2),k)

R =
   1.0000 - 0.8660i
   1.0000 + 0.8660i
```

```
  -4.0000 + 0.0000i
p =
   0.4500 + 0.7794i
   0.4500 - 0.7794i
  -0.8000 + 0.0000i
k =
     []
b1 =
    2.0000    0.4500
a1 =
    1.0000   -0.9000    0.8100
```

Hence

$$X(z) = \frac{(2) + (0.45)z^{-1}}{1 + (-0.9)z^{-1} + (0.81)z^{-2}} + \frac{(-4)}{1 - (-0.8)z^{-1}}$$

2. Computation of the causal sequence $x(n)$ from the $X(z)$ so that it contains no complex numbers: Matlab Script:

```
%% P0414b.m
[Ac,As,r,v0] = inv_CC_PP(b1(1),b1(2),a1(2),a1(3));
disp(sprintf('\nx1(n)
= %2.0f*(%3.1f)^n*cos(%5.4f*pi*n)u(n) ',Ac,r,v0));
disp(sprintf('
+ %5.4f*(%3.1f)^n*sin(%5.4f*pi*n)u(n)\n',As,r,v0));

x1(n) =  2*(0.9)^n*cos(0.3333*pi*n)u(n)
 + 1.7321*(0.9)^n*sin(0.3333*pi*n)u(n)
```

Hence the sequence $x(n)$ is:

$$x(n) = 2(0.9)^n \cos(\pi n/3)u(n) + \sqrt{3}(0.9)^n \sin(\pi n/3)u(n) - 4(-0.8)^n u(n)$$

P4.15

For the linear and time-invariant systems described by the following impulse responses, determine (i) the system function representation, (ii) the difference equation representation, (iii) the pole-zero plot, and (iv) the output $y(n)$ if the input is $x(n) = \left(\frac{1}{4}\right)^n u(n)$.

1. $h(n) = 5(1/4)^n u(n)$
2. $h(n) = n(1/3)^n u(n) + (-1/4)^n u(n)$
3. $h(n) = 3(0.9)^n \cos(\pi n/4 + \pi/3)u(n+1)$
4. $h(n) = \dfrac{(0.5)^n \sin[(n+1)\pi/3]}{\sin(\pi/3)} u(n)$
5. $h(n) = [2 - \sin(\pi n)]u(n)$

Solutions

1. $h(n) = 5(1/4)^n u(n)$

i. The system function: Taking the z-transform of $h(n)$,

$$H(z) = \mathcal{Z}[h(n)] = \mathcal{Z}\left[5(1/4)^n u(n)\right] = \frac{5}{1-0.25z^{-1}}, \ |z| > 0.5$$

ii. The difference equation representation: From $H(z)$ above,

$$y(n) = 5x(n) + 0.25y(n-1)$$

iii. The pole-zero plot is shown in Figure 4.6.

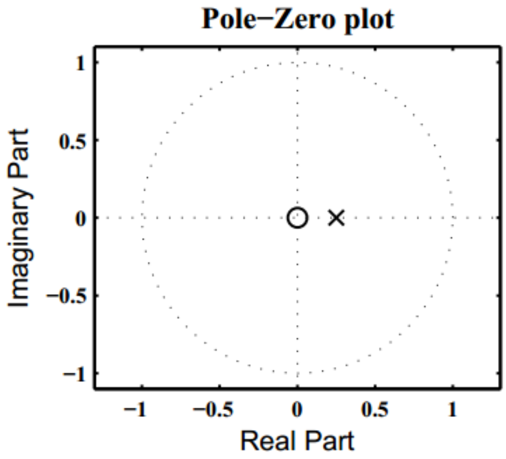


Figure 4.6: Problem P4.15.1 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = \left(\frac{1}{4}\right)^n u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[(1/4)^n u(n)\right] = \frac{1}{1-0.25z^{-1}}, \ |z| > 0.25$$

Now the z-transform of $y(n)$ is

$$Y(z) = H(z)X(z) = \left(\frac{5}{1-0.25z^{-1}}\right)\left(\frac{1}{1-0.25z^{-1}}\right) = \frac{5}{\left(1-0.25z^{-1}\right)^2}, |z| > 0.25$$

$$= 20z\frac{0.25z^{-1}}{\left(1-0.25z^{-1}\right)^2}, \ |z| > 0.25$$

Hence

$$y(n) = 20(n+1)(0.25)^{n+1}u(n+1)$$

2. $h(n) = n(1/3)^n u(n) + \ (-1/4)^n u(n)$

i. The system function: Taking the z-transform of $h(n)$

$$
\begin{aligned}
H(z) &= \mathcal{Z}\left[h(n)\right] = \mathcal{Z}\left[n(1/3)^n u(n) + (-1/4)^n u(n)\right] \\
&= \frac{(1/3)z^{-1}}{\left[1-(1/3)z^{-1}\right]^2} + \frac{1}{1+(1/4)z^{-1}}, \ |z| > (1/3) \\
&= \frac{1-\frac{1}{3}z^{-1}+\frac{7}{36}z^{-2}}{1-\frac{5}{12}z^{-1}-\frac{1}{18}z^{-2}+\frac{1}{36}z^{-3}}, \ |z| > (1/3)
\end{aligned}
$$

ii. The difference equation representation: From $H(z)$ above,

$$
y(n) = x(n) - \frac{1}{3}x(n-1) + \frac{7}{36}x(n-2) + \frac{5}{12}y(n-1) + \frac{1}{18}y(n-2) - \frac{1}{36}y(n-3)
$$

iii. The pole-zero plot is shown in Figure 4.7.

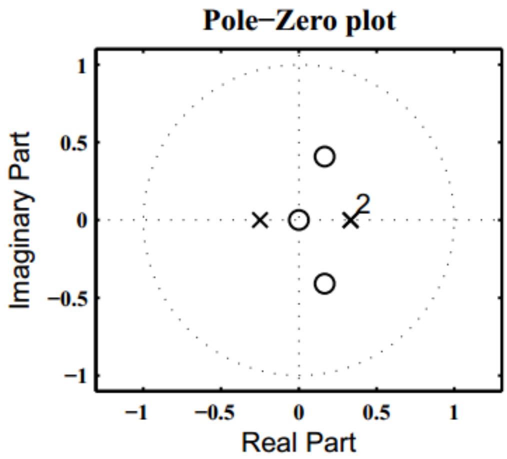


Figure 4.7: Problem P4.15.2 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = \left(\frac{1}{4}\right)^n u(n)$: Taking the z-transform of $x(n)$,

$$
X(z) = \mathcal{Z}\left[(1/4)^n u(n)\right] = \frac{1}{1-\frac{1}{4}z^{-1}}, \ |z| > \frac{1}{4}
$$

Now the z-transform of $y(n)$ is

$$
\begin{aligned}
Y(z) = H(z)X(z) &= \left(\frac{1-\frac{1}{3}z^{-1}+\frac{7}{36}z^{-2}}{1-\frac{5}{12}z^{-1}-\frac{1}{18}z^{-2}+\frac{1}{36}z^{-3}}\right)\left(\frac{1}{1-0.25z^{-1}}\right), |z| > \frac{1}{3} \\
&= \frac{-16}{1-\frac{1}{3}z^{-1}} + \frac{4}{\left(1-\frac{1}{3}z^{-1}\right)^2} + \frac{\frac{1}{2}}{1+\frac{1}{4}z^{-1}} + \frac{\frac{25}{2}}{1-\frac{1}{4}z^{-1}}, |z| > \frac{1}{3} \\
&= \frac{-16}{1-\frac{1}{3}z^{-1}} + 12z\frac{\frac{1}{3}z^{-1}}{\left(1-\frac{1}{3}z^{-1}\right)^2} + \frac{\frac{1}{2}}{1+\frac{1}{4}z^{-1}} + \frac{\frac{25}{2}}{1-\frac{1}{4}z^{-1}}, |z| > \frac{1}{3}
\end{aligned}
$$

Hence

$$
y(n) = -16(\tfrac{1}{3})^n u(n) + 12(n+1)(\tfrac{1}{3})^{n+1}u(n+1) + \frac{1}{2}(-\tfrac{1}{4})^n u(n) + \frac{25}{2}(\tfrac{1}{4})^n u(n)
$$

3. $h(n) = 3(0.9)^n \cos(\pi n/4 + \pi/3)u(n+1)$: Consider

$$
\begin{aligned}
h(n) &= \frac{10}{3}\left[(0.9)^{n+1}\cos\left\{\frac{\pi(n+1)}{4}+\frac{\pi}{12}\right\}u(n+1)\right] \\
&= \left[\frac{10}{3}\cos\left(\frac{\pi}{12}\right)\right](0.9)^{n+1}\cos\left[\frac{\pi}{4}(n+1)\right]u(n+1) - \left[\frac{10}{3}\sin\left(\frac{\pi}{12}\right)\right](0.9)^{n+1}\sin\left[\frac{\pi}{4}(n+1)\right]u(n+1) \\
&= 3.2198(0.9)^{n+1}\cos\left[\frac{\pi}{4}(n+1)\right]u(n+1) - 0.8627(0.9)^{n+1}\sin\left[\frac{\pi}{4}(n+1)\right]u(n+1)
\end{aligned}
$$

i. The system function: Taking the z-transform of $h(n)$

$$
\begin{aligned}
H(z) = \mathcal{Z}[h(n)] &= z\left(3.2198\frac{1-0.6364z^{-1}}{1-1.2728z^{-1}+0.81z^{-2}} - 0.8627\frac{0.6364z^{-1}}{1-1.2728z^{-1}+0.81z^{-2}}\right) \\
&= \frac{3.2198z-2.5981}{1-1.2728z^{-1}+0.81z^{-2}}, \ |z|>0.9
\end{aligned}
$$

ii. The difference equation representation: From $H(z)$ above,

$$
y(n) = 3.2198x(n+1) - 2.5981x(n) + 1.2728y(n-1) - 0.81y(n-2)
$$

iii. The pole-zero plot is shown in Figure 4.8.

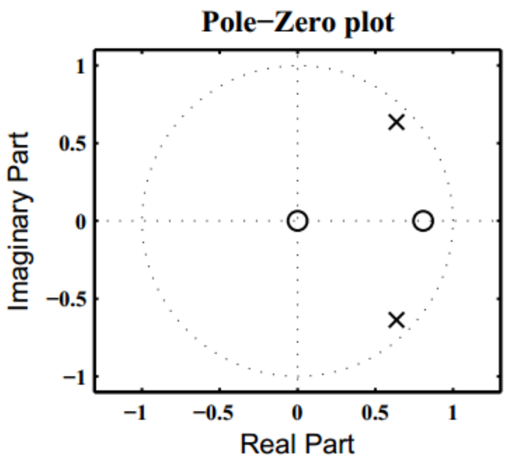


Figure 4.8: Problem P4.15.3 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = \left(\frac{1}{4}\right)^n u(n)$: The z-transform of $x(n)$ is $X(z) = \frac{1}{1-0.25z^{-1}}$, $|z| > 0.25$. Now the z-transform of $y(n)$ is

$$
\begin{aligned}
Y(z) = H(z)X(z) &= \left(\frac{3.2198z-2.5981}{1-1.2728z^{-1}+0.81z^{-2}}\right)\left(\frac{1}{1-0.25z^{-1}}\right), \ |z|>0.9 \\
&= z\left(\frac{4.0285-2.6203z^{-1}}{1.0000-1.2728z^{-1}+0.81z^{-2}} - \frac{0.8087}{1-\frac{1}{4}z^{-1}}\right), \ |z|>0.9
\end{aligned}
$$

Hence

$$
y(n) = \left\{4.0285(0.9)^{n+1}\cos\left[\frac{\pi(n+1)}{4}\right] - 0.0889(0.9)^{n+1}\sin\left[\frac{\pi(n+1)}{4}\right] - 0.8087(\tfrac{1}{4})^{n+1}\right\}u(n+1)
$$

4. $h(n) = \dfrac{(0.5)^n \sin[(n+1)\pi/3]}{\sin(\pi/3)}u(n)$: Consider

$$h(n)=\frac{1}{\sin(\frac{\pi}{3})}\left[(0.5)^n \sin\left\{\frac{\pi n}{3}+\frac{\pi}{3}\right\}u(n)\right]$$
$$=\left[\frac{1}{\sin(\frac{\pi}{3})}\sin\left(\frac{\pi}{3}\right)\right](0.5)^n\sin\left[\frac{\pi}{3}n\right]u(n)+\left[\frac{1}{\sin(\frac{\pi}{3})}\cos\left(\frac{\pi}{3}\right)\right](0.5)^n\cos\left[\frac{\pi}{3}n\right]u(n)$$
$$=(0.5)^n\sin\left[\frac{\pi}{3}n\right]u(n)+0.5774(0.5)^n\cos\left[\frac{\pi}{3}n\right]u(n)$$

i. The system function: Taking the z-transform of $h(n)$

$$H(z)=\mathcal{Z}[h(n)]=\frac{0.4330z^{-1}}{1-0.5z^{-1}+0.25z^{-2}}+0.5774\frac{1-0.25z^{-1}}{1-0.5z^{-1}+0.25z^{-2}}$$
$$=\frac{0.5774+0.2887z^{-1}}{1-0.5z^{-1}+0.25z^{-2}},\ |z|>0.5$$

ii. The difference equation representation: From $H(z)$ above,

$$y(n)=0.5774x(n)+0.2887x(n-1)+0.5y(n-1)-0.25y(n-2)$$

iii. The pole-zero plot is shown in Figure 4.9.

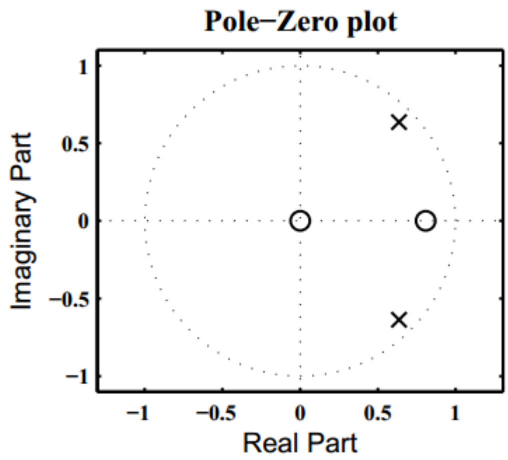


Figure 4.9: Problem P4.15.4 pole-zero plot

iv. The output $y(n)$ for the input $x(n)=\left(\frac{1}{4}\right)^n u(n)$: The z-transform of $x(n)$ is $X(z)=\frac{1}{1-0.25z^{-1}}$, $|z|>0.25$. Now the z-transform of $y(n)$ is

$$Y(z)=H(z)X(z)=\left(\frac{0.5774+0.2887z^{-1}}{1-0.5z^{-1}+0.25z^{-2}}\right)\left(\frac{1}{1-0.25z^{-1}}\right),\ |z|>0.5$$
$$=\frac{0.5774z^{-1}}{1.0000-0.5z^{-1}+0.25z^{-2}}-\frac{0.5774}{1-\frac{1}{4}z^{-1}},\ |z|>0.5$$

Hence

$$y(n)=\frac{4}{3}(0.5)^n\sin(\tfrac{\pi}{3}n)u(n)+0.5774(\tfrac{1}{4})^n u(n)$$

5. $h(n)=[2-\sin(\pi n)]u(n)=2u(n)$

i. The system function: $H(z)=Z[h(n)]=Z[2u(n)]=\dfrac{2}{1-z^{-1}},|z|>1$.

ii. The difference equation representation: $y(n) = 2x(n) + y(n - 1)$
iii. The pole-zero plot is shown in Figure 4.10.

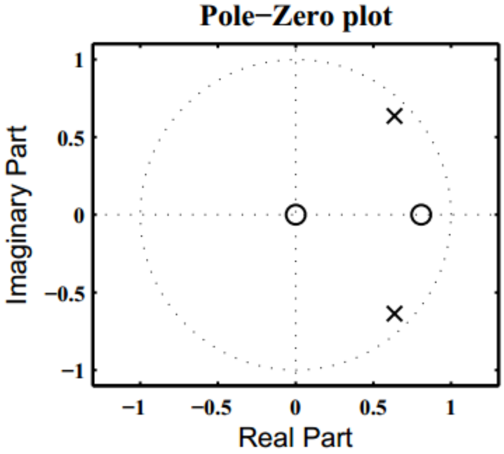


Figure 4.10: Problem P4.15.5 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = \left(\frac{1}{4}\right)^n u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[(1/4)^n u(n)\right] = \frac{1}{1 - 0.25z^{-1}}, \ |z| > 0.25$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{2}{1 - z^{-1}}\right)\left(\frac{1}{1 - 0.25z^{-1}}\right), \ |z| > 1 \\ &= \frac{8/3}{1 - z^{-1}} - \frac{2/3}{1 - \frac{1}{4}z^{-1}}, \ |z| > 1 \end{aligned}$$

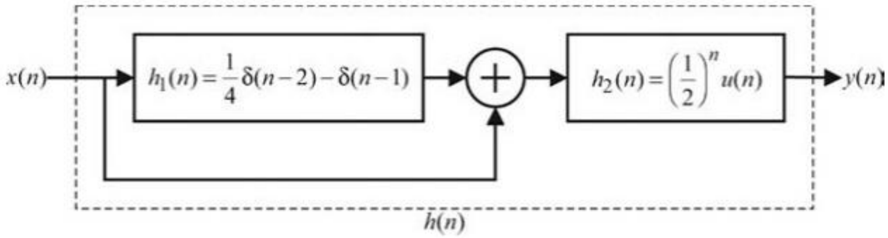
Hence

$$y(n) = \frac{8}{3}u(n) - \frac{2}{3}\left(\frac{1}{4}\right)^n u(n)$$

P4.16

Consider the system shown below.
1. Using the z-transform approach, show that the impulse response, $h(n)$, of the overall system is given by

$$h(n) = \delta(n) - \frac{1}{2}\delta(n - 1)$$



2. Determine the difference equation representation of the overall system that relates the output $y(n)$ to the input $x(n)$.
3. Is this system causal? BIBO stable? Explain clearly to receive full credit.
4. Determine the frequency response $H(e^{jw})$ of the overall system.
5. Using MATLAB, provide a plot of this frequency response over $0 \le \omega \le \pi$.

Solutions

1. The overall system impulse response, $h(n)$, using the z-transform approach: The above system is given by

$$H(z) = H_2(z)\,[1 + H_1(z)] = \frac{1}{1-0.5z^{-1}}\left[1 + 0.25z^{-2} - z^{-1}\right]$$
$$= \frac{\left(1-0.5z^{-1}\right)^2}{1-0.5z^{-1}} = 1 - 0.5z^{-1},\ |z| \neq 0$$

Hence after taking inverse z-transform , we obtain

$$h(n) = \delta(n) - \frac{1}{2}\delta(n-1)$$

2. Difference equation representation of the overall system: From the overall system function $H(z)$,

$$H(z) = \frac{Y(z)}{X(z)} = 1 - 0.5z^{-1} \Rightarrow y(n) = x(n) - 0.5x(n-1)$$

3. Causality and stability: Since $h(n) = 0$ for $n < 0$, the system is causal. Since $h(n)$ is of finite duration (only two samples), $h(n)$ is absolutely summable. Hence BIBO stable.
4. Frequency response $H(e^{jw})$ of the overall system.

$$H\left(e^{j\omega}\right) = \mathcal{F}\left[h(n)\right] = \mathcal{F}\left[\delta(n) - \frac{1}{2}\delta(n-1)\right] = 1 - \frac{1}{2}e^{-j\omega}$$

5. Frequency response over $0 \le \omega \le \pi$ is shown in Figure 4.11.

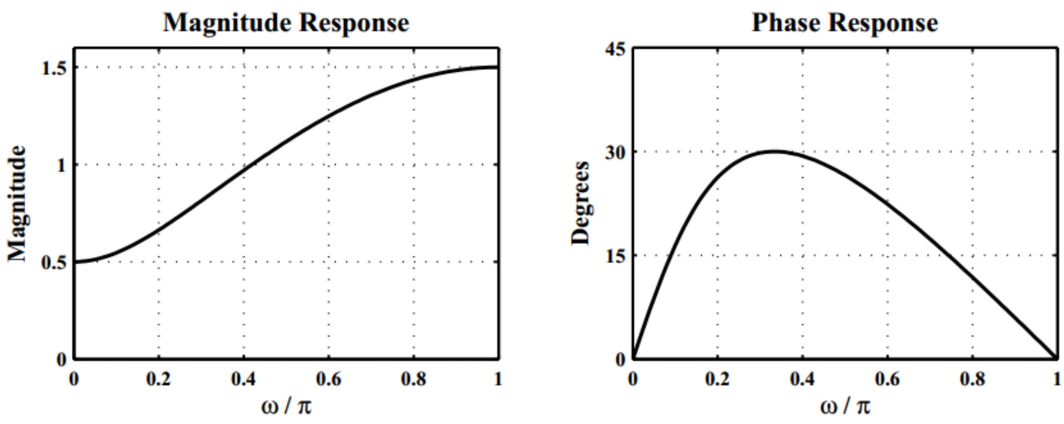


Figure 4.11: Problem P4.16 frequency-response plot

P4.17

For the linear and time-invariant systems described by the following system functions, determine (i) the impulse response representation, (ii) the difference equation representation, (iii) the pole-zero plot, and (iv) the output $y(n)$ if the input is $x(n) = 3\cos(\pi n/3)u(n)$.

1. $H(z) = (z+1)/(z-0.5)$, causal system
2. $H(z) = (1+z^{-1}+z^{-2})/(1+0.5z^{-1}-0.25z^{-2})$, stable system
3. $H(z) = (z^2-1)/(z-3)^2$, anticausal system
4. $H(z) = \dfrac{z}{z-0.25} + \dfrac{1-0.5z^{-1}}{1+2z^{-1}}$, stable system
5. $H(z) = (1+z^{-1}+z^{-2})^2$

Solutions

1. $H(z) = (z + 1)/(z - 0.5)$, causal system. Consider

$$H(z) = \frac{z+1}{z-0.5} = \frac{1+z^{-1}}{1-0.5z^{-1}} = \frac{1}{1-0.5z^{-1}} + \frac{z^{-1}}{1-0.5z^{-1}}, \ |z| > 0.5$$

i. The impulse response: Taking the inverse z-transform of $H(z)$,

$$h(n) = \mathcal{Z}^{-1}[H(z)] = (0.5)^n u(n) + (0.5)^{n-1} u(n-1)$$

ii. The difference equation representation: From $H(z)$ above,

$$y(n) = x(n) + x(n-1) + 0.5y(n-1)$$

iii. The pole-zero plot is shown in Figure 4.12.

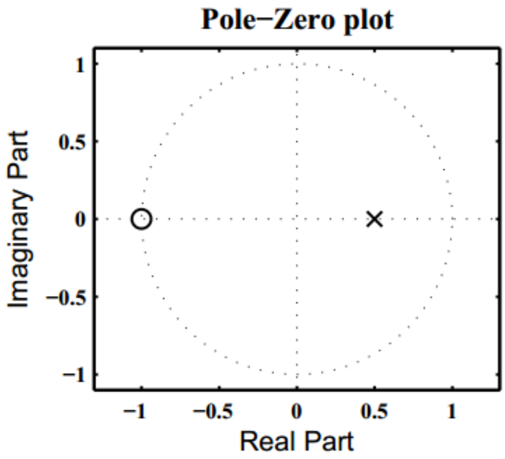


Figure 4.12: Problem P4.17.1 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 3\cos(\pi n/3)u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}[3\cos(\pi n/3)u(n)] = 3\frac{1-[\cos(\pi/3)]z^{-1}}{1-[2\cos(\pi/3)]z^{-1}+z^{-2}} = 3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}}, \ |z|>1$$

Now the z-transform of $y(n)$ is

$$Y(z) = H(z)X(z) = \left(\frac{1+z^{-1}}{1-0.5z^{-1}}\right)\left(3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}}\right) = 3\frac{1+z^{-1}}{1-z^{-1}+z^{-2}}, \ |z|>1$$

$$= 3\frac{1-0.5z^{-1}+1.5z^{-1}}{1-z^{-1}+z^{-2}} = 3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}} + 3\sqrt{3}\frac{\frac{\sqrt{3}}{2}z^{-1}}{1-z^{-1}+z^{-2}}, \ |z|>1$$

Hence

$$y(n) = 3\cos(\pi n/3)u(n) + 3\sqrt{3}\sin(\pi n/3)u(n)$$

2. $H(z) = (1 + z^{-1} + z^{-2})/(1 + 0.5z^{-1} - 0.25z^{-2})$, stable system. Consider

$$H(z) = \frac{1+z^{-1}+z^{-2}}{1+0.5z^{-1}-0.25z^{-2}} = -4 + \frac{0.9348}{1+0.809z^{-1}} + \frac{4.0652}{1-0.309z^{-1}}, \ |z|>0.809$$

i. The impulse response: Taking the inverse z-transform of $H(z)$,

$$h(n) = \mathcal{Z}^{-1}[H(z)] = -4\delta(n) + 0.9348(-0.809)^n u(n) + 4.0652(0.309)^n u(n)$$

ii. The difference equation representation: From $H(z)$ above,

$$y(n) = x(n) + x(n-1) + x(n-2) - 0.5y(n-1) + 0.25y(n-2)$$

iii. The pole-zero plot is shown in Figure 4.13.

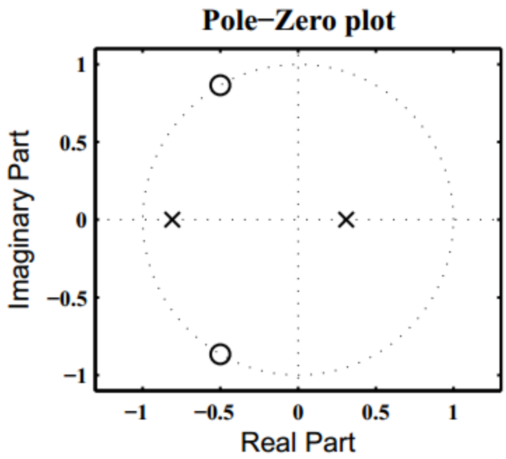


Figure 4.13: Problem P4.17.2 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 3\cos(\pi n/3)u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[3\cos(\pi n/3)u(n)\right] = 3\frac{1-[\cos(\pi/3)]z^{-1}}{1-[2\cos(\pi/3)]z^{-1}+z^{-2}} = 3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}},\ |z|>1$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{1+z^{-1}+z^{-2}}{1+0.5z^{-1}-0.25z^{-2}}\right)\left(3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}}\right),\ |z|>1 \\ &= \frac{1.2055}{1+0.809z^{-1}} - \frac{0.9152}{1-0.309z^{-1}} + \frac{2.7097\left(1-\frac{1}{2}z^{-1}\right)}{1-z^{-1}+z^{-2}} + \frac{3.3524\left(\frac{\sqrt{3}}{2}z^{-1}\right)}{1-z^{-1}+z^{-2}},\ |z|>1 \end{aligned}$$

Hence

$$\begin{aligned} y(n) = {} & 1.2055(-0.809)^n u(n) - 0.9152(0.309)^n u(n) + 2.7097\cos(\pi n/3)u(n) \\ & + 3.3524\sin(\pi n/3)u(n) \end{aligned}$$

3.$H(z) = (z^2 - 1)/(z - 3)^2$, anti-causal system. Consider

$$H(z) = \frac{z^2-1}{(z-3)^2} = \frac{1-z^{-2}}{1-6z^{-1}+9z^{-2}} = -\frac{1}{9} + \frac{2/9}{1-3z^{-1}} + \frac{8z}{27}\frac{3z^{-1}}{\left(1-3z^{-1}\right)^2},\ |z|<3$$

i. The impulse response: Taking the inverse z-transform of $H(z)$,

$$h(n) = \mathcal{Z}^{-1}\left[H(z)\right] = -\frac{1}{9}\delta(n) - \frac{2}{9}3^n u(-n-1) - \frac{8}{27}(n+1)3^{n+1}u(-n-2)$$

ii. The difference equation representation: From $H(z)$ above,

$$y(n) = x(n) - x(n-2) + 6y(n-1) - 9y(n-2)$$

iii. The pole-zero plot is shown in Figure 4.14.

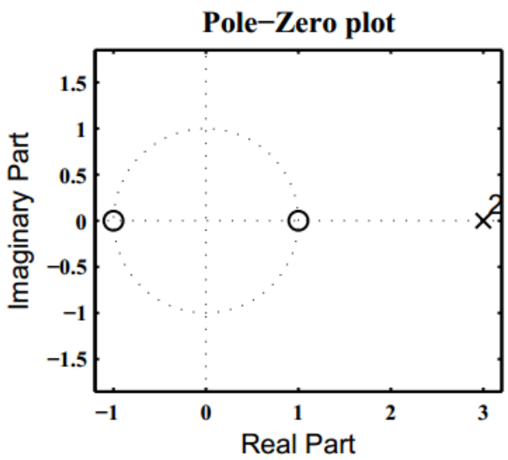


Figure 4.14: Problem P4.17.3 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 3\cos(\pi n/3)u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[3\cos(\pi n/3)u(n)\right] = 3\frac{1-[\cos(\pi/3)]z^{-1}}{1-[2\cos(\pi/3)]z^{-1}+z^{-2}} = 3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}},\ |z|>1$$

Now the z-transform of $y(n)$ is

$$Y(z) = H(z)X(z) = \left(\frac{1-z^{-2}}{1-6z^{-1}+9z^{-2}}\right)\left(3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}}\right),\ 1<|z|<3$$

$$= \frac{43/49}{1-3z^{-1}} + \frac{20z}{21}\frac{3z^{-1}}{\left(1-3z^{-1}\right)^2} + \frac{\frac{-36}{49}\left(1-\frac{1}{2}z^{-1}\right)}{1-z^{-1}+z^{-2}} + \frac{\frac{193}{1820}\left(\frac{\sqrt{3}}{2}z^{-1}\right)}{1-z^{-1}+z^{-2}},\ |z|>1$$

Hence

$$y(n) = -\frac{43}{49}3^n u(-n-1) - \frac{20}{21}(n+1)3^{n+1}u(-n-2) - \frac{36}{49}\cos(\pi n/3)u(n) + \frac{193}{1820}\sin(\pi n/3)u(n)$$

4. $H(z) = \dfrac{z}{z-0.25} + \dfrac{1-0.5z^{-1}}{1+2z^{-1}}$, stable system. Consider

$$H(z) = \frac{1}{1-0.25z^{-1}} + \frac{1-0.5z^{-1}}{1+2z^{-1}} = \frac{2+\frac{5}{4}z^{-1}+\frac{1}{8}z^{-2}}{1+\frac{7}{4}z^{-1}-\frac{1}{2}z^{-2}}$$

$$= -\frac{1}{4} + \frac{1}{1-0.25z^{-1}} + \frac{5/4}{1+2z^{-1}},\ 0.25<|z|<2$$

i. The impulse response: Taking the inverse z-transform of $H(z)$,

$$h(n) = \mathcal{Z}^{-1}\left[H(z)\right] = -\frac{1}{4}\delta(n) + \left(\frac{1}{4}\right)^n u(n) - \frac{5}{4}2^n u(-n-1)$$

ii. The difference equation representation: From $H(z)$ above,

$$y(n) = 2x(n) + \frac{5}{4}x(n-1) - \frac{1}{8}x(n-2) - \frac{7}{4}y(n-1) + \frac{1}{2}y(n-2)$$

iii. The pole-zero plot is shown in Figure 4.15.

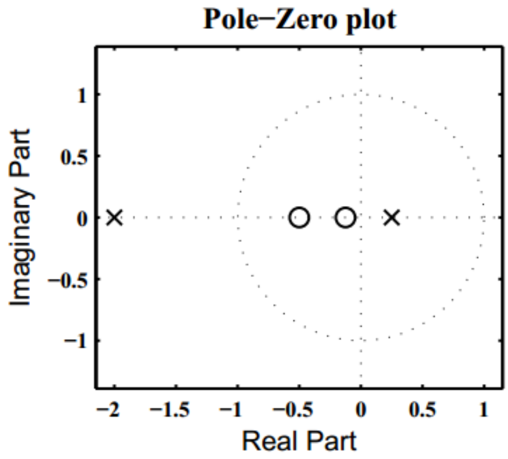


Figure 4.15: Problem P4.17.4 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 3\cos(\pi n/3)u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}[3\cos(\pi n/3)u(n)] = 3\frac{1-[\cos(\pi/3)]z^{-1}}{1-[2\cos(\pi/3)]z^{-1}+z^{-2}} = 3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}},\ |z|>1$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{2+\frac{5}{4}z^{-1}+\frac{1}{8}z^{-2}}{1+\frac{7}{4}z^{-1}-\frac{1}{2}z^{-2}}\right)\left(3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}}\right),\ 1<|z|<3 \\ &= \frac{75/28}{1+2z^{-1}} - \frac{3/13}{1-\frac{1}{4}z^{-1}} + \frac{\frac{1293}{364}\left(1-\frac{1}{2}z^{-1}\right)}{1-z^{-1}+z^{-2}} - \frac{\frac{323}{2553}\left(\frac{\sqrt{3}}{2}z^{-1}\right)}{1-z^{-1}+z^{-2}},\ |z|>1 \end{aligned}$$

Hence

$$\begin{aligned} y(n) = &-\frac{75}{28}2^n u(-n-1) - \frac{3}{13}\left(\frac{1}{4}\right)^n u(n) + \frac{1293}{364}\cos(\pi n/3)u(n) \\ &- \frac{323}{2553}\sin(\pi n/3)u(n) \end{aligned}$$

5. $H(z) = (1 + z^{-1} + z^{-2})^2$. Consider

$$H(z) = (1+z^{-1}+z^{-2})^2 = 1+2z^{-1}+3z^{-2}+2z^{-3}+z^{-4},\ |z|>0$$

i. The impulse response: Taking the inverse z-transform of $H(z)$,

$$h(n) = \mathcal{Z}^{-1}[H(z)] = \{\underset{\uparrow}{1}, 2, 3, 2, 1\}$$

ii. The difference equation representation: From $H(z)$ above,

$$y(n) = x(n) + 2x(n-1) + 3x(n-2) + 2x(n-3) + x(n-4)$$

iii. The pole-zero plot is shown in Figure 4.16.

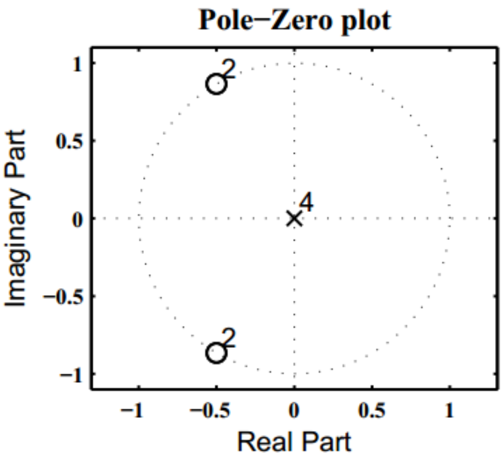


Figure 4.16: Problem P4.17.5 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 3\cos(\pi n/3)u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[3\cos(\pi n/3)u(n)\right] = 3\frac{1-[\cos(\pi/3)]z^{-1}}{1-[2\cos(\pi/3)]z^{-1}+z^{-2}} = 3\frac{1-0.5z^{-1}}{1-z^{-1}+z^{-2}},\ |z|>1$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= 3\left[\frac{\left(1+2z^{-1}+3z^{-2}+2z^{-3}+z^{-4}\right)\left(1-0.5z^{-1}\right)}{1-z^{-1}+z^{-2}}\right],\ |z|>1 \\ &= 9+\frac{3}{2}z^{-1}-\frac{3}{2}z^{-2}-\frac{3}{2}z^{-3}+\frac{\frac{1293}{364}\left(1-\frac{1}{2}z^{-1}\right)}{1-z^{-1}+z^{-2}}-\frac{\frac{328}{2553}\left(\frac{\sqrt{3}}{2}z^{-1}\right)}{1-z^{-1}+z^{-2}},\ |z|>1 \end{aligned}$$

Hence

$$\begin{aligned} y(n) = 9\delta(n) &+ \frac{3}{2}\delta(n-1)-\frac{3}{2}\delta(n-2)-\frac{3}{2}\delta(n-3)+\frac{1293}{364}\cos(\pi n/3)u(n) \\ &-\frac{328}{2553}\sin(\pi n/3)u(n) \end{aligned}$$

P4.18

For the linear, causal, and time-invariant systems described by the following difference equations, determine (i) the impulse response representation, (ii) the system function representation, (iii) the pole-zero plot, and (iv) the output $y(n)$ if the input is $x(n) = 2(0.9)^n u(n)$.

1. $y(n) = [x(n) + 2x(n-1) + x(n-3)]/4$
2. $y(n) = x(n) + 0.5x(n-1) - 0.5y(n-1) + 0.25y(n-2)$
3. $y(n) = 2x(n) + 0.9y(n-1)$
4. $y(n) = -0.45x(n) - 0.4x(n-1) + x(n-2) + 0.4y(n-1) + 0.45y(n-2)$
5. $y(n) = \sum_{m=0}^{4}(0.8)^m x(n-m) - \sum_{\ell=1}^{4}(0.9)^\ell y(n-\ell)$

Solutions

1. $y(n) = [x(n) + 2x(n - 1) + x(n - 3)]/4$

i. The system function representation: Taking the z-transform of the above difference equation,

$$Y(z) = \frac{1}{4}[X(z) + 2z^{-1}X(z) + z^{-3}X(z)] \Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{1 + 2z^{-1} + z^{-3}}{4}$$

ii. The impulse response: Taking the inverse z-transform of $H(z)$,

$$h(n) = \frac{1}{4}[\delta(n) + 2\delta(n-1) + \delta(n-3)]$$

iii. The pole-zero plot is shown in Figure 4.17.

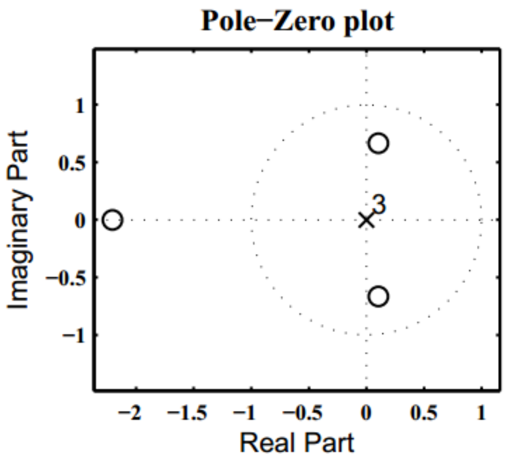


Figure 4.17: Problem P4.18.1 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 2(0.9)^n u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[2(0.9)^n u(n)\right] = \frac{2}{1 - 0.9z^{-1}}, \ |z| > 0.9$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{1 + 2z^{-1} + z^{-3}}{4}\right)\left(\frac{2}{1 - 0.9z^{-1}}\right), \ |z| > 0.9 \\ &= -\frac{1310}{729} - \frac{50}{81}z^{-1} - \frac{5}{9}z^{-2} + \frac{\frac{990}{431}}{1 - 0.9z^{-1}}, \ |z| > 0.9 \end{aligned}$$

Hence

$$y(n) = -\frac{1310}{729}\delta(n) - \frac{50}{81}\delta(n-1) - \frac{5}{9}\delta(n-2) + \frac{990}{431}(0.9)^n u(n)$$

2. $y(n) = x(n) + 0.5x(n - 1) - 0.5y(n - 1) + 0.25y(n - 2)$

i. The system function representation: Taking the z-transform of the above difference equation,

$$Y(z) = X(z) + 0.5z^{-1}X(z) - 0.5z^{-1}Y(z) + 0.25z^{-2}Y(z)$$

or

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1 + 0.5z^{-1}}{1 + 0.5z^{-1} - 0.25z^{-2}} = \frac{1 + 0.5z^{-1}}{\left(1 + 0.809z^{-1}\right)\left(1 - 0.309z^{-1}\right)}, \; |z| > 0.809$$

ii. The impulse response: Taking the inverse z-transform of $H(z)$,

$$\begin{aligned} h(n) &= \mathcal{Z}^{-1}\left[\frac{1 + 0.5z^{-1}}{\left(1 + 0.809z^{-1}\right)\left(1 - 0.309z^{-1}\right)}\right] = \mathcal{Z}^{-1}\left[\frac{0.2764}{1 + 0.809z^{-1}} + \frac{0.7236}{1 - 0.309z^{-1}}\right] \\ &= 0.2764(-0.809)^n u(n) + 0.7236(0.308)^n u(n) \end{aligned}$$

iii. The pole-zero plot is shown in Figure 4.18.

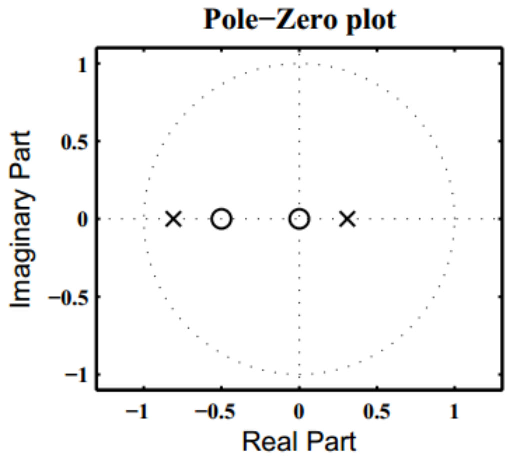


Figure 4.18: Problem P4.18.2 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 2(0.9)^n u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[2(0.9)^n u(n)\right] = \frac{2}{1 - 0.9z^{-1}}, \; |z| > 0.9$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{1 + 0.5z^{-1}}{1 + 0.5z^{-1} - 0.25z^{-2}}\right)\left(\frac{2}{1 - 0.9z^{-1}}\right), \; |z| > 0.9 \\ &= \frac{0.2617}{1 + 0.809z^{-1}} - \frac{0.7567}{1 - 0.309z^{-1}} + \frac{2.495}{1 - 0.9z^{-1}}, \; |z| > 0.9 \end{aligned}$$

Hence

$$y(n) = 0.2617(-0.809)^n u(n) - 0.7567(0.309)^n u(n) + 2.495(0.9)^n u(n)$$

3. $y(n) = 2x(n) + 0.9y(n - 1)$

i. The system function representation: Taking the z-transform of the above difference equation,

$$Y(z) = 2X(z) + 0.9z^{-1}Y(z)$$

or

$$H(z) = \frac{Y(z)}{X(z)} = \frac{2}{1 - 0.9z^{-1}}, \; |z| > 0.9$$

ii. The impulse response: Taking the inverse z-transform of $H(z)$,

$$h(n) = 2(0.9)^n u(n)$$

iii. The pole-zero plot is shown in Figure 4.19.

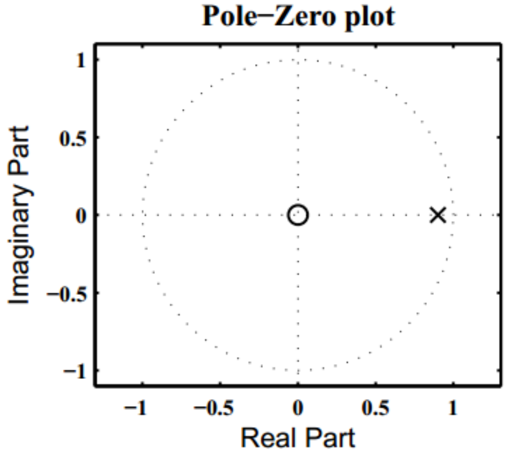


Figure 4.19: Problem P4.18.3 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 2(0.9)^n u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[2(0.9)^n u(n)\right] = \frac{2}{1 - 0.9z^{-1}}, \ |z| > 0.9$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{2}{1 - 0.9z^{-1}}\right)\left(\frac{2}{1 - 0.9z^{-1}}\right), |z| > 0.9 \\ &= \frac{40}{9} z \frac{0.9z^{-1}}{\left(1 - 0.9z^{-1}\right)^2}, \ |z| > 0.9 \end{aligned}$$

Hence

$$y(n) = \frac{40}{9}(n+1)(0.9)^{n+1} u(n+1)$$

4. $y(n) = -0.45x(n) - 0.4x(n-1) + x(n-2) + 0.4y(n-1) + 0.45y(n-2)$

i. The system function representation: Taking the z-transform of the above difference equation,

$$Y(z) = -0.45X(z) - 0.4z^{-1}X(z) + z^{-2}X(z) + 0.4z^{-1}Y(z) + 0.45z^{-2}Y(z)$$

or

$$H(z) = \frac{Y(z)}{X(z)} = \frac{-0.45 - 0.4z^{-1} + z^{-2}}{1 - 0.4z^{-1} - 0.45z^{-2}} = \frac{-0.45 - 0.4z^{-1} + z^{-2}}{\left(1 + 0.5z^{-1}\right)\left(1 - 0.9z^{-1}\right)}, \ |z| > 0.9$$

ii. The impulse response: Taking the inverse z-transform of $H(z)$,

$$\begin{aligned} h(n) &= \mathcal{Z}^{-1}\left[\frac{-0.45 - 0.4z^{-1} + z^{-2}}{\left(1 + 0.5z^{-1}\right)\left(1 - 0.9z^{-1}\right)}\right] = \mathcal{Z}^{-1}\left[-\frac{20}{9} + \frac{1.5536}{1 + 0.5z^{-1}} + \frac{0.2187}{1 - 0.9z^{-1}}\right] \\ &= -\frac{20}{9}\delta(n) + 1.5536(-0.5)^n u(n) + 0.2187(0.9)^n u(n) \end{aligned}$$

iii. The pole-zero plot is shown in Figure 4.20.

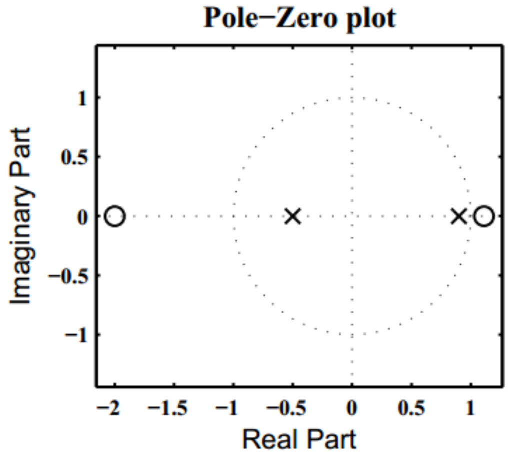


Figure 4.20: Problem P4.18.4 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 2(0.9)^n u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[2(0.9)^n u(n)\right] = \frac{2}{1-0.9z^{-1}}, \ |z| > 0.9$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{-0.45-0.4z^{-1}+z^{-2}}{1-0.4z^{-1}-0.45z^{-2}}\right)\left(\frac{2}{1-0.9z^{-1}}\right), \ |z| > 0.9 \\ &= \frac{1.1097}{1+0.5z^{-1}} - \frac{2.4470}{1-0.9z^{-1}} + 0.4859z\frac{0.9z^{-1}}{\left(1-0.9z^{-1}\right)^2}, \ |z| > 0.9 \end{aligned}$$

Hence

$$y(n) = 1.1097(-0.5)^n u(n) - 2.4470(0.9)^n u(n) + 0.4859(n+1)(0.9)^{n+1} u(n+1)$$

5. $y(n) = \sum_{m=0}^{4} (0.8)^m x(n-m) - \sum_{l=1}^{4} (0.9)^l y(n-l)$

i. The system function representation: Taking the z-transform of the above difference equation,

$$Y(z) = \sum_{m=0}^{4} (0.8)^m z^{-m} X(z) - \sum_{\ell=1}^{4} (0.9)^\ell z^{-\ell} Y(z)$$

or

$$\begin{aligned} H(z) = \frac{Y(z)}{X(z)} &= \frac{\sum_{m=0}^{4} (0.8)^m z^{-m}}{1+\sum_{\ell=1}^{4} (0.9)^\ell z^{-\ell}} \\ &= \frac{1+0.8z^{-1}+0.64+z^{-2}+0.512z^{-3}+0.4096z^{-4}}{\left(1-0.5562z^{-1}+0.81z^{-2}\right)\left(1+1.4562z^{-1}+0.81z^{-2}\right)}, \ |z| > 0.9 \end{aligned}$$

ii. The impulse response: Taking the inverse z-transform of $H(z)$,

$$\begin{aligned} h(n) &= \mathcal{Z}^{-1}\left[0.6243 + \frac{0.1873+0.0651z^{-1}}{1-0.5562z^{-1}+0.81z^{-2}} + \frac{0.1884+0.1353z^{-1}}{1+1.4562z^{-1}+0.81z^{-2}}\right] \\ &= 0.1884\delta(n) + \left[0.1879(0.9)^n \cos(0.4\pi n + 4.63^\circ) + 0.1885(0.9)^n \cos(0.8\pi n + 1.1^\circ)\right] u(n) \end{aligned}$$

iii. The pole-zero plot is shown in Figure 4.21.

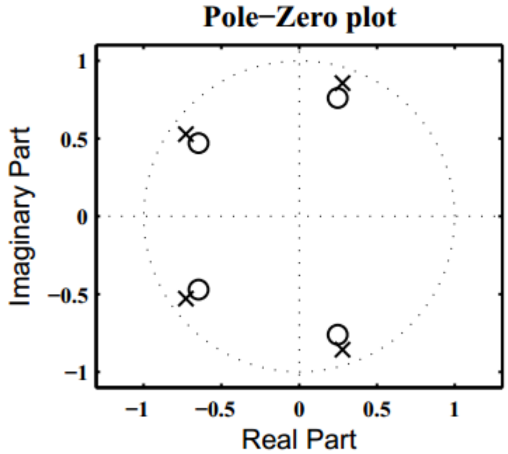


Figure 4.21: Problem P4.18.5 pole-zero plot

iv. The output $y(n)$ for the input $x(n) = 2(0.9)^n u(n)$: Taking the z-transform of $x(n)$,

$$X(z) = \mathcal{Z}\left[2(0.9)^n u(n)\right] = \frac{2}{1 - 0.9z^{-1}}, \ |z| > 0.9$$

Now the z-transform of $y(n)$ is

$$\begin{aligned} Y(z) = H(z)X(z) &= \left(\frac{1 + 0.8z^{-1} + 0.64 + z^{-2} + 0.512z^{-3} + 0.4096z^{-4}}{1 + 0.9z^{-1} + 0.81 + z^{-2} + 0.279z^{-3} + 0.6561z^{-4}}\right)\left(\frac{2}{1 - 0.9z^{-1}}\right) \\ &= \frac{0.2081 + 0.1498z^{-1}}{1 - 0.5562z^{-1} + 0.81z^{-2}} + \frac{0.1896 + 0.1685z^{-1}}{1 + 1.4562z^{-1} + 0.81z^{-2}} + \frac{1.6023}{1 - 0.9z^{-1}}, \ |z| > 0.9 \end{aligned}$$

Hence

$$\begin{aligned} y(n) = {} & 0.3197(0.9)^n \cos(0.4\pi n - 49.37^\circ)u(n) + 0.1982(0.9)^n \cos(0.8\pi n - 16.9^\circ)u(n) \\ & + 1.6023(0.9)^n u(n) \end{aligned}$$

P4.19

The output sequence $y(n)$ in Problem P4.18 is the total response. For each of the systems given in Problem P4.18, separate $y(n)$ into (i) the homogeneous part, (ii) the particular part, (iii) the transient response, and (iv) the steady-state response.

Solutions

1. $y(n) = [x(n) + 2x(n - 1) + x(n - 3)]/4$: The total response is

$$y(n) = -\frac{1310}{729}\delta(n) - \frac{50}{81}\delta(n-1) - \frac{5}{9}\delta(n-2) + \frac{990}{431}(0.9)^n u(n)$$

i. Homogeneous part: Since the system is an FIR filter, the homogeneous equation is $y(n) = 0$. Thus $y_h(n) = 0$.
ii. Particular part: Hence the total response is the particular part, or

$$y_{\mathrm{p}}(n) = -\frac{1310}{729}\delta(n) - \frac{50}{81}\delta(n-1) - \frac{5}{9}\delta(n-2) + \frac{990}{431}(0.9)^n u(n)$$

iii. Transient response: Since the entire response decays to zero,

$$y_{\mathrm{tr}}(n) = -\frac{1310}{729}\delta(n) - \frac{50}{81}\delta(n-1) - \frac{5}{9}\delta(n-2) + \frac{990}{431}(0.9)^n u(n)$$

iv. Steady-state response: Clearly, $y_{ss}(n) = 0$.

2. $y(n) = x(n) + 0.5x(n - 1) - 0.5y(n - 1) + 0.25y(n - 2)$: The total response is

$$y(n) = 0.2617(-0.809)^n u(n) - 0.7567(0.309)^n u(n) + 2.495(0.9)^n u(n)$$

i. Homogeneous part: The first two terms in $y(n)$ are due to the system poles, hence

$$y_{\mathrm{h}}(n) = 0.2617(-0.809)^n u(n) - 0.7567(0.309)^n u(n)$$

ii. Particular part: The last term in $y(n)$ is due to the input pole, hence

$$y_{\mathrm{p}}(n) = 2.495(0.9)^n u(n)$$

iii. Transient response: Since all poles of $Y(z)$ are inside the unit circle,

$$y_{\mathrm{tr}}(n) = 0.2617(-0.809)^n u(n) - 0.7567(0.309)^n u(n) + 2.495(0.9)^n u(n)$$

iv. Steady-state response: Clearly, $y_{ss}(n) = 0$.

3. $y(n) = 2x(n) + 0.9y(n - 1)$: The total response is

$$y(n) = \frac{40}{9}(n+1)(0.9)^{n+1}u(n+1)$$

i. Homogeneous part: Since the system pole and the input pole are the same and hence are indistinguishable. Therefore, the total response can be equally divided into two parts or

$$y_{\mathrm{h}}(n) = \frac{20}{9}(n+1)(0.9)^{n+1}u(n+1)$$

ii. Particular part: Since the system pole and the input pole are the same and hence are indistinguishable. Therefore, the total response can be equally divided into two parts or

$$y_{\mathrm{p}}(n) = \frac{20}{9}(n+1)(0.9)^{n+1}u(n+1)$$

iii. Transient response: Since all poles of $Y(z)$ are inside the unit circle,

$$y_{\mathrm{tr}}(n) = \frac{40}{9}(n+1)(0.9)^{n+1}u(n+1)$$

iv. Steady-state response: Clearly, $y_{ss}(n) = 0$.

4. $y(n) = -0.45x(n) - 0.4x(n - 1) + x(n - 2) + 0.4y(n - 1) + 0.45y(n - 2)$: The total response is

$$y(n) = 1.1097(-0.5)^n u(n) - 2.4470(0.9)^n u(n) + 0.4859(n+1)(0.9)^{n+1}u(n+1)$$

i. Homogeneous part: There are two system poles, $p_1 = -0.5$ and $p_2 = 0.9$. Clearly, p_2 is also an input pole. Hence the response due to p_2 has to be divided to include in both parts. Hence

$$y_{\mathrm{h}}(n) = 1.1097(-0.5)^n u(n) - 1.1135(0.9)^n u(n) + 0.24295(n+1)(0.9)^{n+1}u(n+1)$$

ii. Particular part: from above,

$$y_p(n) = -1.1135(0.9)^n u(n) + 0.24295(n+1)(0.9)^{n+1} u(n+1)$$

iii. Transient response: Since all poles of $Y(z)$ are inside the unit circle,

$$y_{tr}(n) = 1.1097(-0.5)^n u(n) - 2.4470(0.9)^n u(n) + 0.4859(n+1)(0.9)^{n+1} u(n+1)$$

iv. Steady-state response: Clearly, $y_{ss}(n) = 0$.

5. $y(n) = \sum_{m=0}^{4}(0.8)^m x(n-m) - \sum_{l=1}^{4}(0.9)^l y(n-l)$: The total response is

$$y(n) = 0.3197(0.9)^n \cos(0.4\pi n - 49.37°)u(n) + 0.1982(0.9)^n \cos(0.8\pi n - 16.9°)u(n) + 1.6023(0.9)^n u(n)$$

i. Homogeneous part: The first two terms in $y(n)$ are due to the system poles, hence

$$y_h(n) = 0.3197(0.9)^n \cos(0.4\pi n - 49.37°)u(n) + 0.1982(0.9)^n \cos(0.8\pi n - 16.9°)u(n)$$

ii. Particular part: The last term in $y(n)$ is due to the input pole, hence

$$y_p(n) = 1.6023(0.9)^n u(n)$$

iii. Transient response: Since all poles of $Y(z)$ are inside the unit circle,

$$y_{tr}(n) = 0.3197(0.9)^n \cos(0.4\pi n - 49.37°)u(n) + 0.1982(0.9)^n \cos(0.8\pi n - 16.9°)u(n) + 1.6023(0.9)^n u(n)$$

iv. Steady-state response: Clearly, $y_{ss}(n) = 0$.

P4.20

A stable system has four zeros and four poles as given here:

$$zeros: \pm 1, \pm j1 \qquad poles: \pm 0.9, \pm j0.9$$

It is also known that the frequency response function $H(e^{jw})$ evaluated at $\omega = \pi/4$ is equa to 1, i.e.,

$$H(e^{j\pi/4}) = 1$$

1. Determine the system function $H(z)$, and indicate its region of convergence.
2. Determine the difference equation representation.
3. Determine the steady-state response $y_{ss}(n)$ if the input is $x(n) = \cos(\pi n/4)u(n)$.
4. Determine the transient response $y_{tr}(n)$ if the input is $x(n) = \cos(\pi n/4)u(n)$.

Solutions

It is also known that $H(e^{j\pi/4}) = 1$.

1. The system function $H(z)$ and its region of convergence: Consider

$$H(z) = K\frac{(z-1)(z+1)(z-j)(z+j)}{(z-0.9)(z+0.9)(z-j0.9)(z+j0/9)} = K\frac{1-z^{-4}}{1-0.6561z^{-4}},\ |z| > 0.9$$

Now at z = $e^{j\pi/4}$, we have $H(e^{j\pi/4}) = 1$. Hence

$$1 = H\left(e^{j\pi/4}\right) = K\frac{1-e^{j\pi}}{1-0.6561e^{j\pi}} = K \times 1.2077 \Rightarrow K = 0.8281$$

or

$$H(z) = \frac{0.8281\left(1-z^{-4}\right)}{1-0.6561z^{-4}},\ |z| > 0.9$$

2. The difference equation representation: From

$$H(z) = \frac{0.8281\left(1-z^{-4}\right)}{1-0.6561z^{-4}} = \frac{Y(z)}{X(z)}$$

we have

$$y(n) = 0.8281x(n) - 0.8281x(n-4) + 0.6561y(n-4)$$

3. The steady-state response $y_{ss}(n)$ for the input $x(n) = \cos(\pi n/4)u(n)$: From the z-transform table,

$$X(z) = \frac{1-[\cos(\pi/4)]z^{-1}}{1-[2\cos(\pi/4)]z^{-1}+z^{-2}} = \frac{1-\frac{1}{\sqrt{2}}z^{-1}}{1-\sqrt{2}z^{-1}+z^{-2}}$$

Hence

$$\begin{aligned} Y(z) = H(z)X(z) &= \left[\frac{0.8281\left(1-z^{-4}\right)}{1-0.6561z^{-4}}\right]\left(\frac{1-\frac{1}{\sqrt{2}}z^{-1}}{1-\sqrt{2}z^{-1}+z^{-2}}\right) \\ &= \frac{1-\frac{1}{\sqrt{2}}z^{-1}}{1-\sqrt{2}z^{-1}+z^{-2}} - \frac{0.0351}{1-0.9z^{-1}} - \frac{0.0509}{1+0.9z^{-1}} + \frac{-0.0860-0.1358z^{-1}}{1-0.81z^{-2}},\ |z| > 1 \end{aligned}$$

The first term above has poles on the unit circle and hence gives the steady-state response

$$y_{ss}(n) = \cos(\pi n/4)$$

4. The transient response $y_{tr}(n)$ for the input $x(n) = \cos(\pi n/4)u(n)$: The remaining terms in $y(n)$ are the transient response terms. Using the **inv_CC_PP** function we have

$$Y_{tr}(z) = -\frac{0.0351}{1-0.9z^{-1}} - \frac{0.0509}{1+0.9z^{-1}} - 0.0860\frac{1}{1-0.81z^{-2}} - 0.1509\frac{0.9z^{-1}}{1-0.81z^{-2}},\ |z| > 1$$

Hence

$$\begin{aligned} y_{tr}(n) = &-0.0351(0.9)^n u(n) - 0.0509(-0.9)^n u(n) - 0.086(0.9)^n \cos(\pi n/2)u(n) \\ &- 0.1509(0.9)^n \sin(\pi n/2)u(n) \end{aligned}$$

P4.21

A digital filter is described by the frequency response function

$$H(e^{j\omega}) = [1 + 2\cos(\omega) + 3\cos(2\omega)]\cos\left(\frac{\omega}{2}\right)e^{-j5\omega/2}$$

1. Determine the difference equation representation.
2. Using the **freqz** function, plot the magnitude and phase of the frequency response of the filter. Note the magnitude and phase at $\omega = \pi/2$ and at $\omega = \pi$.
3. Generate 200 samples of the signal $x(n) = \sin(\pi n/2) + 5\cos(\pi n)$, and process through the filter to obtain $y(n)$. Compare the steady-state portion of $y(n)$ to $x(n)$. How are the amplitudes and phases of two sinusoids affected by the filter?

Solutions

A digital filter is described by the frequency response function

$$H(e^{j\omega}) = [1 + 2\cos(\omega) + 3\cos(2\omega)]\cos\left(\frac{\omega}{2}\right)e^{-j5\omega/2}$$

which can be written as

$$\begin{aligned} H(e^{j\omega}) &= \left[1 + 2\left(\frac{e^{j\omega} + e^{-j\omega}}{2}\right) + 3\left(\frac{e^{j2\omega} + e^{-j2\omega}}{2}\right)\right]\frac{e^{j\frac{1}{2}\omega} + e^{-j\frac{1}{2}\omega}}{2}e^{-j\frac{5}{2}\omega} \\ &= \frac{3}{4} + \frac{5}{4}e^{-j\omega} + e^{-j2\omega} + e^{-j3\omega} + \frac{5}{4}e^{-j4\omega} + \frac{3}{4}e^{-j5\omega} \end{aligned}$$

or after substituting $e^{-jw} = z^{-1}$, we obtain

$$H(z) = \frac{3}{4} + \frac{5}{4}z^{-1} + z^{-2} + z^{-3} + \frac{5}{4}z^{-4} + \frac{3}{4}z^{-5}$$

1. The difference equation representation: From $H(z)$ above

$$y(n) = \frac{3}{4}x(n) + \frac{5}{4}x(n-1) + x(n-2) + x(n-3) + \frac{5}{4}x(n-4) + \frac{3}{4}x(n-5)$$

2. The magnitude and phase response plots are shown in Figure 4.22.

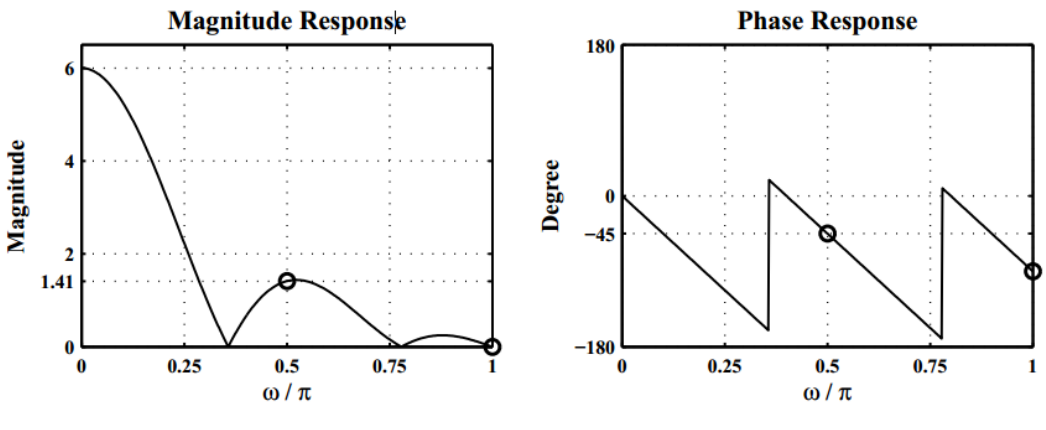


Figure 4.22: Problem P4.21.2 frequency-response plots

The magnitude and phase at ω = π/2 are $\sqrt{2}$ and -45°, respectively. The magnitude at ω = π is zero.

3. The output sequence $y(n)$ for the input $x(n) = \sin(\pi n/2) + 5\cos(\pi n)$: Matlab script:

```
% P4.21
clc; close all;
b = [3/4 5/4 1 1 5/4 3/4]; a = [1];
n = 0:200; x = sin(pi*n/2)+5*cos(pi*n); y =
filter(b,a,x);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0421c');
subplot(2,1,1); Hs = stem(n,x); set(Hs,'markersize',2);
axis([-2 202 -7 6]);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
title('x(n) = sin(\pi \times n / 2)+5 \times cos(\pi
\times n)',...
'FontSize',12);
subplot(2,1,2); Hs = stem(n,y); set(Hs,'markersize',2);
axis([-2 202 -2 4]);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
title('Output sequence after filtering','FontSize',12);
print -deps2 ../epsfiles/P0421c;
```

The input and output sequence plots are shown in Figure 4.23. It shows that the sinusoidal sequence with the input frequency ω = π is completely suppressed in the steady-state output. The steady-state response of $x(n) = sin(\pi n/2)$ should be (using the magnitude and phase at ω = π/2 computed in part 2. above)

$$\begin{aligned} y_{ss}(n) &= \sqrt{2}\sin(\pi n/2 - 45^\circ) = \sqrt{2}\cos(45^\circ)\sin(\pi n/2) - \sqrt{2}\sin(45^\circ)\cos(\pi n/2) \\ &= sin(\pi n/2) - cos(\pi n/2) = \{\ldots, \underset{\uparrow}{-1}, 1, 1, -1, -1, \ldots\} \end{aligned}$$

as verified in the bottom plot of Figure 4.23.

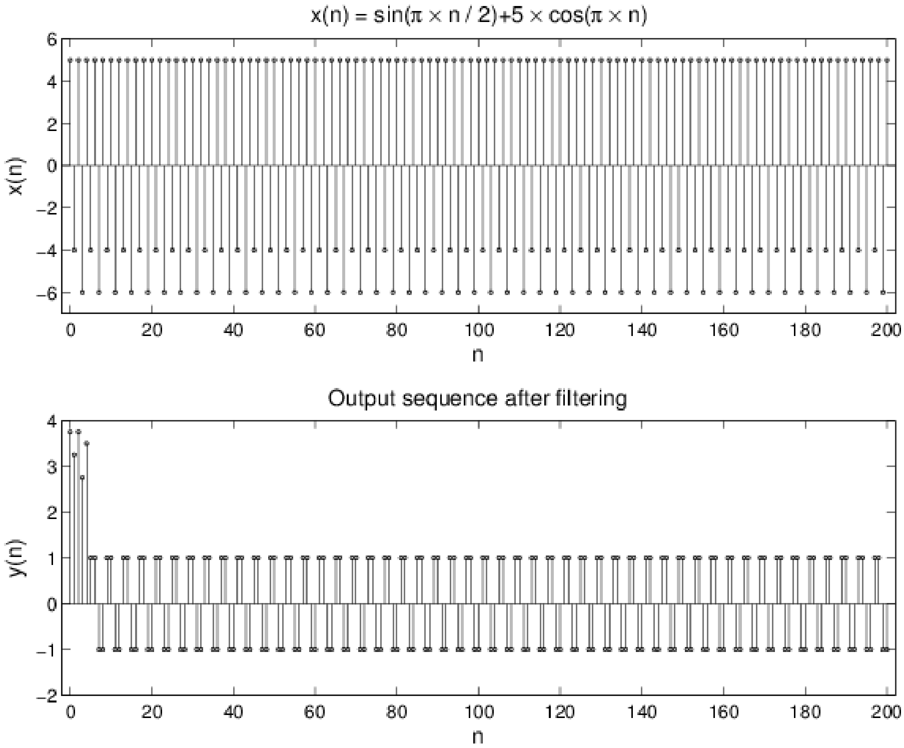


Figure 4.23: Problem P4.21.3 input and output sequence plots

P4.22

Repeat Problem 4.21 for the following filter

$$H(e^{j\omega}) = \frac{1 + e^{-j4\omega}}{1 - 0.8145e^{-j4\omega}}$$

Solutions

A digital filter is described by the frequency response function

$$H(e^{j\omega}) = \frac{1 + e^{-j4\omega}}{1 - 0.8145e^{-j4\omega}}$$

which after substituting $e^{-j\omega} = z^{-1}$ becomes

$$H(z) = \frac{1 + z^{-4}}{1 - 0.8145z^{-4}}$$

1. The difference equation representation: From $H(z)$ above

$$y(n) = x(n) + x(n-4) + 0.8145y(n-4)$$

2. The magnitude and phase response plots are shown in Figure 4.24.

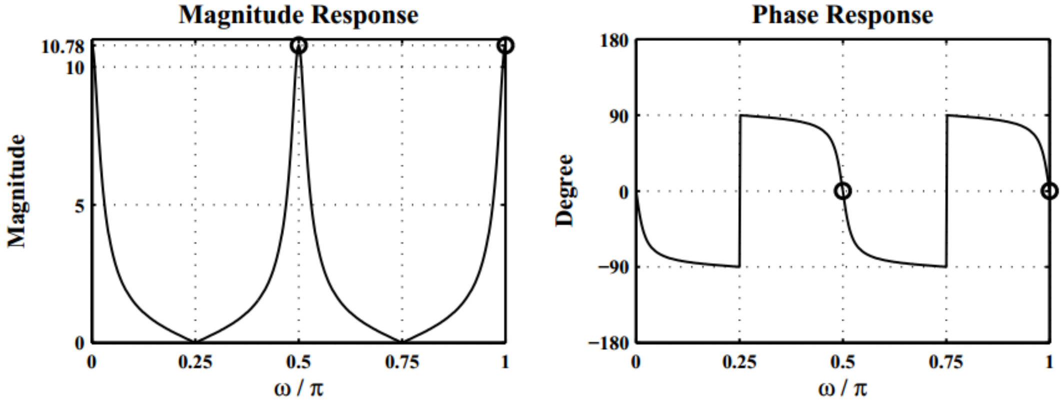


Figure 4.24: Problem P4.22.2 frequency-response plots

The magnitudes and phases at both $\omega = \pi/2$ and $\omega = \pi$ are 10.78 and 0°, respectively.

3. The output sequence $y(n)$ for the input $x(n) = \sin(\pi n/2) + 5\cos(\pi n)$: Matlab script:

```
% P4.22
clc; close all;
b = [1 0 0 0 1]; a = [1 0 0 0 -0.8145];
n = 0:200; x = sin(pi*n/2)+5*cos(pi*n); y =
filter(b,a,x);
Hf_1 = figure;
set(Hf_1,'NumberTitle','off','Name','P0422c');
subplot(2,1,1); Hs = stem(n,x); set(Hs,'markersize',2);
axis([-2 202 -7 7]);
xlabel('n','FontSize',12); ylabel('x(n)','FontSize',12);
title('x(n) = sin(\pi \times n / 2)+5 \times cos(\pi
\times n)',...
'FontSize',12);
subplot(2,1,2); Hs = stem(n,y); set(Hs,'markersize',2);
axis([-2 202 -70 70]);
xlabel('n','FontSize',12); ylabel('y(n)','FontSize',12);
title('Output sequence after filtering','FontSize',12);
print -deps2 ../epsfiles/P0422c;
```

The input and output sequence plots are shown in Figure 4.25.The steady-state response of $x(n)$ should be (using the magnitude and phase at $\omega = \pi/2$ computed in part 2. above)

$$\begin{aligned} y_{ss}(n) &= 10.78\sin(\pi n/2) + 10.78 \times 5\cos(\pi n) \\ &= 10.78 sin(\pi n/2) + 53.91 cos(\pi n/2) \end{aligned}$$

The bottom plot of Figure 4.25 shows that both sinusoidal sequences have the scaling of 10.78 and no delay distortion.

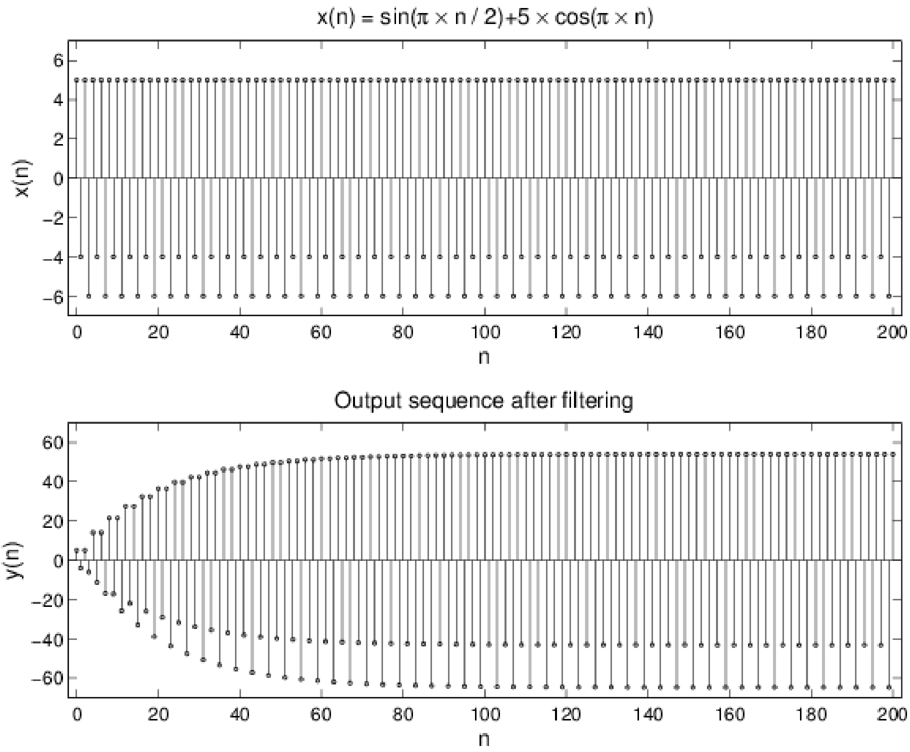


Figure 4.25: Problem P4.22.3 input and output sequence plots

P4.23

Solve the following difference equation for $y(n)$ using the one-sided z-transform approach.

$$\begin{aligned} y(n) &= 0.81y(n-2) + x(n) - x(n-1), \; n \geq 0; \quad y(-1) = 2, \, y(-2) = 2 \\ x(n) &= (0.7)^n u(n+1) \end{aligned}$$

Generate the first 20 samples of $y(n)$ using MATLAB, and compare them with your answer.

Solutions

Notice that

$$x(n) = (0.7)^n u(n+1) = \begin{cases} (0.7)^{-1}, & n = -1; \\ (0.7)^n u(n), & n \geq 0. \end{cases}$$

After taking the one-sided z-transform of the above difference equation, we obtain

$$\begin{aligned} Y^+(z) &= 0.81 \left[y(-2) + y(-1)z^{-1} + z^{-2}Y^+(z) \right] + X^+(z) - \left[x(-1) + z^{-1}X^+(z) \right] \\ &= 0.81z^{-2}Y^+(z) + \left[1 - z^{-1} \right] X^+(z) + [0.81y(-2) - x(-1)] + 0.81y(-1)z^{-1} \end{aligned}$$

or

$$Y^+(z) = \frac{1 - z^{-1}}{1 - 0.81z^{-1}} X^+(z) + \frac{[0.81y(-2) - x(-1)] + 0.81y(-1)z^{-1}}{1 - 0.81z^{-1}}$$

After substituting the initial conditions and $X^+(z) = \mathcal{Z}\,[0.7^n u(n)] = \dfrac{1}{1 - 0.7z^{-1}}$, we obtain

$$\begin{aligned}Y^+(z) &= \left(\frac{1 - z^{-1}}{1 - 0.81z^{-1}}\right)\left(\frac{1}{1 - 0.7z^{-1}}\right) + \frac{0.1914 + 1.62z^{-1}}{1 - 0.81z^{-1}} \\ &= \frac{1.1914 + 0.4860z^{-1} - 1.1340z^{-2}}{(1 - 0.81z^{-1})(1 - 0.7z^{-1})} = 2 + \frac{0.4642}{1 - 0.81z^{-1}} + \frac{2.7273}{1 - 0.7z^{-1}}\end{aligned}$$

Hence upon inverse transformation

$$y(n) = 2\delta(n) + 0.4642(0.81)^n u(n) + 2.7273(0.7)^n u(n)$$

Matlab verification:

```
% P4.23
clc; close all;
b1 = [1 -1]; nb1 = [0 1]; a11 = [1 0 -0.81]; na11 = [0 1
2]; a12 = [1 -0.7];
na12 = [0 1]; [a1,na1] = conv_m(a11,na11,a12,na12);
b2 = [0.1914 1.62]; nb2 = [0 1]; a2 = [1 0 -0.81]; na2 =
[0 1 2];
[bnr1,nbnr1] = conv_m(b1,nb1,a2,na2); [bnr2,nbnr2] =
conv_m(b2,nb2,a1,na1);
[b,nb] = sigadd(bnr1,nbnr1,bnr2,nbnr2); [a,na] =
conv_m(a1,na1,a2,na2);
[R,p,k] = residuez(b,a)
n = [0:20]; x = 0.7.^n; xic = [0.1914 1.62];
yb1 = filter(b1,a11,x,xic);
yb2 = R(1)*((p(1)).^n)+R(3)*((p(3)).^n)+R(5)*((p(5)).^n);
error = max(abs(yb1-yb2))

R =
    0.7457
   -0.0000
   -0.2106
    0.0000
    0.6563
p =
    0.9000
    0.9000
   -0.9000
   -0.9000
    0.7000
k =
```

```
     []
error =
   9.7603e-08
```

Chapter 5

P5.1

Compute the DFS coefficients of the following periodic sequences using the DFS definition, and then verify your answers using MATLAB.

1. $\tilde{x}_1(n) = \{4, 1, -1, 1\}$, $N = 4$
2. $\tilde{x}_2(n) = \{2, 0, 0, 0, -1, 0, 0, 0\}$, $N = 8$
3. $\tilde{x}_3(n) = \{1, 0, -1, -1, 0\}$, $N = 5$
4. $\tilde{x}_4(n) = \{0, 0, 2j, 0, 2j, 0\}$, $N = 6$
5. $\tilde{x}_5(n) = \{3, 2, 1\}$, $N = 3$

Solutions

```
% P5.1
%% P0501a
xtilde1 = [4,1,-1,1]; N = 4; Xtilde1 = dfs(xtilde1,N)

Xtilde1 =
   5.0000 + 0.0000i   5.0000 + 0.0000i   1.0000 - 0.0000i
5.0000 + 0.0000i
%% P0501b
xtilde2 = [2,0,0,0,-1,0,0,0]; N = 8; Xtilde2 =
dfs(xtilde2,N)

Xtilde2 =
  Columns 1 through 4
   1.0000 + 0.0000i   3.0000 + 0.0000i   1.0000 - 0.0000i
3.0000 + 0.0000i
  Columns 5 through 8
   1.0000 - 0.0000i   3.0000 + 0.0000i   1.0000 - 0.0000i
3.0000 + 0.0000i
%% P0501c
xtilde3 = [1,0,-1,-1,0]; N = 5; Xtilde3 = dfs(xtilde3,N)

Xtilde3 =
  Columns 1 through 4
  -1.0000 + 0.0000i   2.6180 + 0.0000i   0.3820 - 0.0000i
0.3820 + 0.0000i
  Column 5
   2.6180 + 0.0000i
%% P0501d
xtilde4 = [0,0,2j,0,2j,0]; N = 6; Xtilde4 =
dfs(xtilde4,N)

Xtilde4 =
  Columns 1 through 4
   0.0000 + 4.0000i   0.0000 - 2.0000i   0.0000 - 2.0000i
-0.0000 + 4.0000i
```

```
  Columns 5 through 6
   0.0000 - 2.0000i   0.0000 - 2.0000i
%% P0501e
xtilde5 = [3,2,1]; N = 3; Xtilde5 = dfs(xtilde5,N)

Xtilde5 =
   6.0000 + 0.0000i   1.5000 - 0.8660i   1.5000 + 0.8660i
```

P5.2

Determine the periodic sequences given the following periodic DFS coefficients. First use the IDFS definition and then verify your answers using MATLAB.

1. $\tilde{X}_1(k) = \{4, 3j, -3j\}$, $N = 3$
2. $\tilde{X}_2(k) = \{j, 2j, 3j, 4j\}$, $N = 4$
3. $\tilde{X}_3(k) = \{1, 2+3j, 4, 2-3j\}$, $N = 4$
4. $\tilde{X}_4(k) = \{0, 0, 2, 0, 0\}$, $N = 5$
5. $\tilde{X}_5(k) = \{3, 0, 0, 0, -3, 0, 0, 0\}$, $N = 8$

Solutions

```
% P5.2
%% P0502a
Xtilde1 = [4,1i*3,-1i*3]; N = 3; xtilde1 =
idfs(Xtilde1,N)

xtilde1 =
   1.3333 + 0.0000i  -0.3987 + 0.0000i   3.0654 - 0.0000i
%% P0502b
Xtilde2 = [1i,1i*2,1i*3,1i*4]; N = 4; xtilde2 =
idfs(Xtilde2,N)

xtilde2 =
   0.0000 + 2.5000i   0.5000 - 0.5000i  -0.0000 - 0.5000i
-0.5000 - 0.5000i
%% P0502c
Xtilde3 = [1,2+1i*3,4,2-1i*3]; N = 4; xtilde3 =
idfs(Xtilde3,N)

xtilde3 =
   2.2500 + 0.0000i  -2.2500 + 0.0000i   0.2500 + 0.0000i
0.7500 - 0.0000i
%% P0502d
```

```
Xtilde4 = [0,0,2,0,0]; N = 5; xtilde4 = idfs(Xtilde4,N)

xtilde4 =
  Columns 1 through 4
   0.4000 + 0.0000i  -0.3236 + 0.2351i   0.1236 - 0.3804i
0.1236 + 0.3804i
  Column 5
  -0.3236 - 0.2351i
%% P0502e
Xtilde5 = [3,0,0,0,-3,0,0,0]; N = 8; xtilde5 =
idfs(Xtilde5,N)

xtilde5 =
  Columns 1 through 4
   0.0000 + 0.0000i   0.7500 - 0.0000i   0.0000 + 0.0000i
0.7500 - 0.0000i
  Columns 5 through 8
   0.0000 + 0.0000i   0.7500 - 0.0000i   0.0000 + 0.0000i
0.7500 - 0.0000i
```

P5.3

Let $\tilde{x}_1(n)$ be periodic with fundamental period $N = 40$ where one period is given by

$$\tilde{x}_1(n) = \begin{cases} 5\sin(0.1\pi n), & 0 \le n \le 19 \\ 0, & 20 \le n \le 39 \end{cases}$$

and let $\tilde{x}_2(n)$ be periodic with fundamental period $N = 80$, where one period is given by

$$\tilde{x}_2(n) = \begin{cases} 5\sin(0.1\pi n), & 0 \le n \le 19 \\ 0, & 20 \le n \le 79 \end{cases}$$

These two periodic sequences differ in their periodicity but otherwise have the same nonzero samples.

1. Compute the DFS $\tilde{X}_1(k)$ of $\tilde{x}_1(n)$, and plot samples (using the **stem** function) of its magnitude and angle versus k.

2. Compute the DFS $\tilde{X}_2(k)$ of $\tilde{x}_2(n)$, and plot samples of its magnitude and angle versus k.

3. What is the difference between the two preceding DFS plots?

Solutions

1. Computation of $\tilde{X}_1(k)$ using Matlab:

```
% P5.3
%% P0503a.m
n1 = [0:39]; xtilde1 = [5*sin(0.1*pi*[0:19]),zeros(1,20)];
N1 = length(n1);
[Xtilde1] = dft(xtilde1,N1); k1 = n1;
mag_Xtilde1      =      abs(Xtilde1);      pha_Xtilde1      =
angle(Xtilde1)*180/pi;
zei = find(mag_Xtilde1 < 1000*eps);
pha_Xtilde1(zei) = zeros(1,length(zei));
%                          Hf_1                          =
figure('Units','normalized','position',[0.1,0.1,0.8,0.8],
'paperunits','inches','paperposition',[0,0,6,5]);
Hf_1                                                     =
figure('Units','normalized','position',[0.1,0.1,0.8,0.8],
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.3.1');

subplot(3,1,1);    H_s1    =    stem(n1,xtilde1,'filled');
set(H_s1,'markersize',3);
axis([-1,N1,-6,6]);
title('One    period    of    the    periodic    sequence
xtilde_1(n)','fontsize',10);
ntick = [n1(1):2:n1(N1),N1]'; ylabel('Amplitude');
set(gca,'XTickMode','manual','XTick',ntick,'FontSize',8);

subplot(3,1,2);   H_s2   =   stem(k1,mag_Xtilde1,'filled');
set(H_s2,'markersize',3);
axis([-1,N1,0,max(mag_Xtilde1)+10]);
title('Magnitude      of      Xtilde_1(k)','fontsize',10);
ylabel('Magnitude');
ktick = [k1(1):2:k1(N1),N1]';
set(gca,'XTickMode','manual','XTick',ktick,'FontSize',8);

subplot(3,1,3);   H_s3   =   stem(k1,pha_Xtilde1,'filled');
set(H_s3,'markersize',3);
title('Phase of Xtilde_1(k)','fontsize',10); xlabel('k');
ylabel('Degrees');
ktick = [k1(1):2:k1(N1),N1]'; axis([-1,N1,-200,200]);
set(gca,'XTickMode','manual','XTick',ktick,'FontSize',8);
```

```
set(gca,'YTickMode','manual','YTick',[-180;-90;0;90;180]);
% set(gcf,'paperpositionmode','auto');
print -deps2 ../epsfiles/P0503a;
```

Plots of $\tilde{x}_1(n)$ and $\tilde{X}_1(k)$ are shown in Figure 5.1.

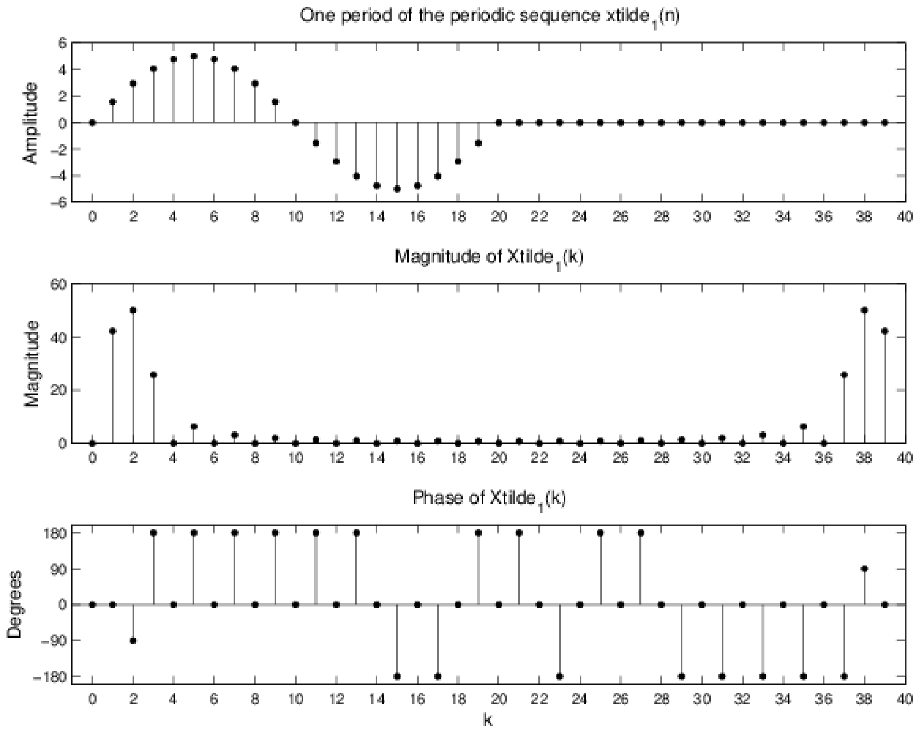


Figure 5.1: Plots of $\tilde{x}_1(n)$ and $\tilde{X}_1(k)$ in Problem 5.3a

2. Computation of $\tilde{X}_2(k)$ using Matlab:

```
%% P0503b.m
n2  =  [0:79];  xtilde2  =  [xtilde1,  zeros(1,40)];  N2  =
length(n2);
[Xtilde2] = dft(xtilde2,N2); k2 = n2;
mag_Xtilde2      =      abs(Xtilde2);      pha_Xtilde2      =
angle(Xtilde2)*180/pi;
zei = find(mag_Xtilde2 < 1000*eps);
pha_Xtilde2(zei) = zeros(1,length(zei));
Hf_2                                                          =
figure('Units','normalized','position',[0.1,0.1,0.8,0.8],
'paperunits','inches');
%                           Hf_2                              =
figure('Units','normalized','position',[0.1,0.1,0.8,0.8],
'paperunits','inches','paperposition',[0,0,6,5]);
%
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,6,5]);
```

```
set(Hf_2,'NumberTitle','off','Name','P5.3.2');

subplot(3,1,1);    H_s1    =    stem(n2,xtilde2,'filled');
set(H_s1,'markersize',3);
title('One    period    of    the    periodic    sequence
xtilde2(n)','fontsize',10);
ntick = [n2(1):5:n2(N2),N2]'; ylabel('xtilde2'); axis([-
1,N2,-6,6]);
set(gca,'XTickMode','manual','XTick',ntick)

subplot(3,1,2);   H_s2   =   stem(k2,mag_Xtilde2,'filled');
set(H_s2,'markersize',3);
axis([-1,N2,0,60]);
title('Magnitude      of      Xtilde2(k)','fontsize',10);
ylabel('|Xtilde2|')
ktick = [k2(1):5:k2(N2),N2]';
set(gca,'XTickMode','manual','XTick',ktick)

subplot(3,1,3);   H_s3   =   stem(k2,pha_Xtilde2,'filled');
set(H_s3,'markersize',3);
title('Phase  of  Xtilde2(k)','fontsize',10);  xlabel('k');
ylabel('Degrees')
ktick = [k2(1):5:k2(N2),N2]'; axis([-1,N2,-200,200]);
set(gca,'XTickMode','manual','XTick',ktick);
set(gca,'YTickMode','manual','YTick',[-180;-90;0;90;180]);
% set(gcf,'paperpositionmode','auto');
print -deps2 ../epsfiles/P0503b;
```

Plots of $\tilde{x}_2(n)$ and $\tilde{X}_2(k)$ are shown in Figure 5.2.

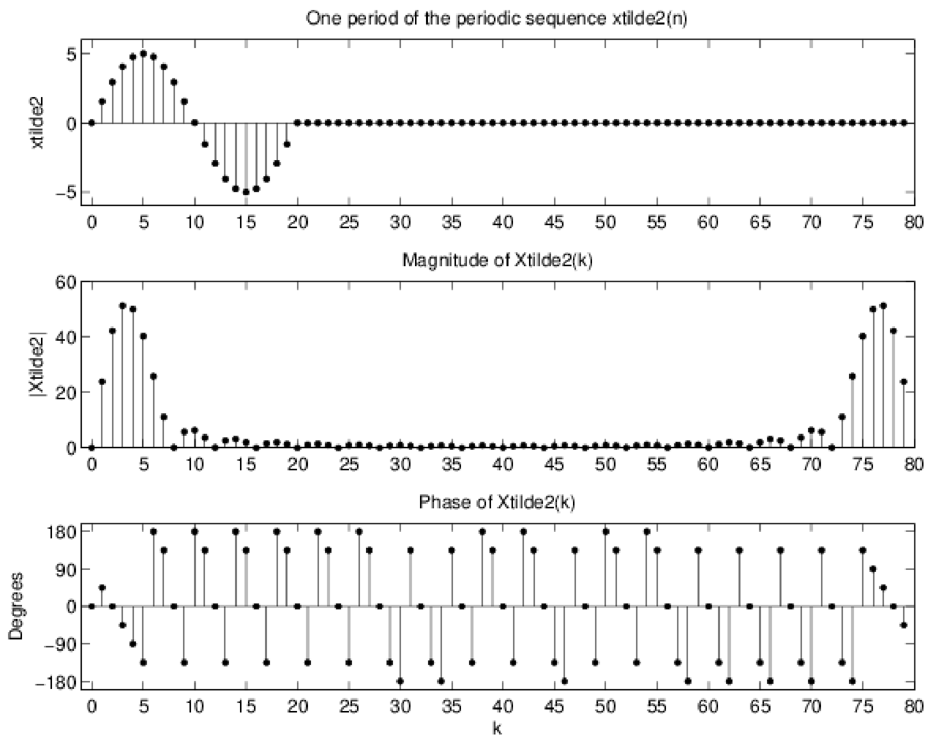


Figure 5.2: Plots of Magnitude and Phase of $\tilde{X}_2(k)$ in Problem 5.3b

P5.4

Consider the periodic sequence $\tilde{x}_1(n)$ given in Problem P5.3. Let $\tilde{x}_2(n)$ be periodic with fundamental period $N = 40$, where one period is given by

$$\tilde{x}_2(n) = \begin{cases} \tilde{x}_1(n), & 0 \le n \le 19 \\ -\tilde{x}_1(n-20), & 20 \le n \le 39 \end{cases}$$

1. Determine analytically the DFS $\tilde{X}_2(k)$ in terms of $\tilde{X}_1(k)$.

2. Compute the DFS $\tilde{X}_2(k)$ of $\tilde{x}_2(n)$ and plot samples of its magnitude and angle versus k.

3. Verify your answer in part 1 using the plots of $\tilde{X}_1(k)$ and $\tilde{X}_2(k)$?

Solutions

1. Determine analytically the DFS $\tilde{X}_2(k)$ in terms of $\tilde{X}_1(k)$.

2. Computation of the DFS $\tilde{X}_2(k)$ using Matlab:

```
% P5.4
n1 = [0:19]; xtilde1 = [5*sin(0.1*pi*n1)];
```

```
n2 = [0:39]; xtilde2 = [xtilde1, -xtilde1]; N2 = length(n2);
[Xtilde2] = dft(xtilde2,N2); k2 = n2;
mag_Xtilde2      =      abs(Xtilde2);      pha_Xtilde2      =
angle(Xtilde2)*180/pi;
zei = find(mag_Xtilde2 < 1000*eps);
pha_Xtilde2(zei) = zeros(1,length(zei));

Hf_1                                                        =
figure('Units','normalized','position',[0.1,0.1,0.8,0.8],
...
'paperunits','inches');
%
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,6,4]);
set(Hf_1,'NumberTitle','off','Name','P5.4.2');

subplot(3,1,1);    H_s1    =    stem(n2,xtilde2,'filled');
set(H_s1,'markersize',3);
axis([-1,N2,-6,6]);
title('One    period    of    the    periodic    sequence
xtilde_2(n)','fontsize',10);
ntick = [n2(1):5:n2(N2),N2]'; ylabel('Amplitude');
set(gca,'XTickMode','manual','XTick',ntick)

subplot(3,1,2);   H_s2   =   stem(k2,mag_Xtilde2,'filled');
set(H_s2,'markersize',3);
axis([-1,N2,0,100]);
title('Magnitude      of      Xtilde2(k)','fontsize',10);
ylabel('Magnitude')
ktick = [k2(1):5:k2(N2),N2]';
set(gca,'XTickMode','manual','XTick',ktick)

subplot(3,1,3);   H_s3   =   stem(k2,pha_Xtilde2,'filled');
set(H_s3,'markersize',3);
title('Phase  of  Xtilde2(k)','fontsize',10);  xlabel('k');
ylabel('Degrees')
ktick = [k2(1):5:k2(N2),N2]'; axis([-1,N2,-200,200]);
set(gca,'XTickMode','manual','XTick',ktick)
set(gca,'YTickMode','manual','YTick',[-180;-90;0;90;180])
print -deps2 ../epsfiles/P0504;
```

Plots of $\tilde{x}_2(n)$ and $\tilde{X}_2(k)$ are shown in Figure 5.3.

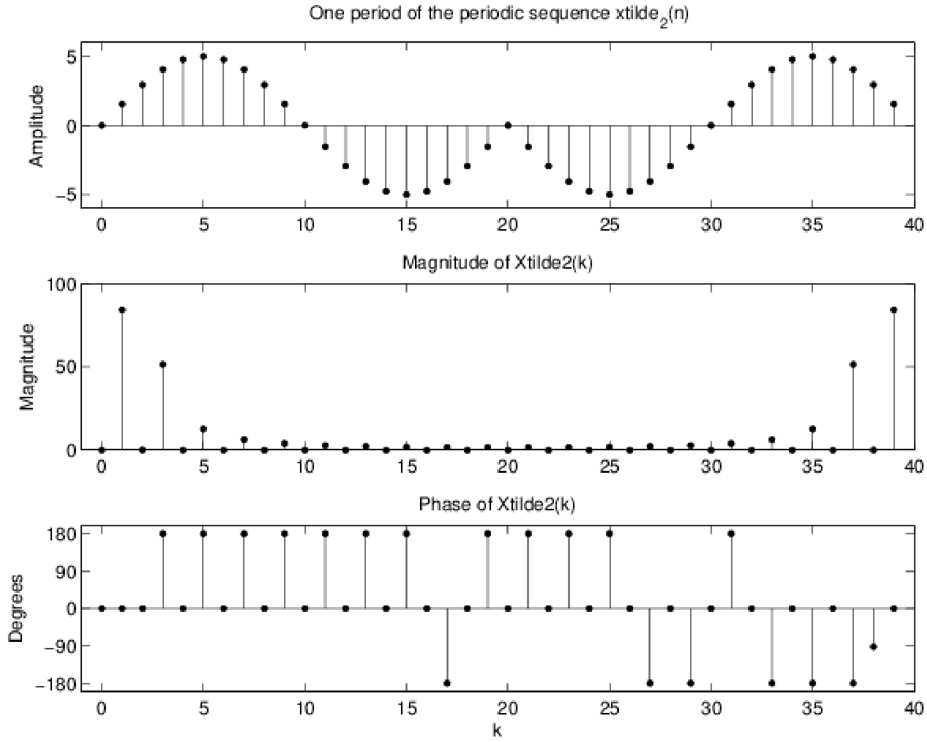


Figure 5.3: Plots of Magnitude and Phase of $\tilde{X}_2(k)$ in Problem 5.4b

3. Verify your answer in part 1 above using the plots of $\tilde{X}_1(k)$ and $\tilde{X}_2(k)$?

P5.5

Consider the periodic sequence $\tilde{x}_1(n)$ given in Problem P5.3. Let $\tilde{x}_3(n)$ be periodic with period 80, obtained by concatenating two periods of $\tilde{x}_1(n)$, i.e.,

$$\tilde{x}_3(n) = [\tilde{x}_1(n), \tilde{x}_1(n)]_{\text{PERIODIC}}$$

Clearly, $\tilde{x}_3(n)$ is different from $\tilde{x}_2(n)$ of Problem P5.3 even though both of them are periodic with period 80.

1. Compute the DFS $\tilde{X}_3(k)$ of $\tilde{x}_3(n)$, and plot samples of its magnitude and angle versus *k*.
2. What effect does the periodicity doubling have on the DFS?
3. Generalize this result to *M*-fold periodicity. In particular, show that if

$$\tilde{x}_M(n) = \Bigg[\underbrace{\tilde{x}_1(n), \tilde{x}_1(n), \ldots, \tilde{x}_1(n)}_{M \text{ times}}\Bigg]_{\text{PERIODIC}}$$

then

$$\begin{array}{lll} \tilde{X}_M(Mk) & = M\tilde{X}_1(k), & k = 0, 1, \ldots, N-1 \\ \tilde{X}_M(k) & = 0, & k \neq 0, M, \ldots, MN \end{array}$$

Solutions

1. Computation and plot of the DFS $\tilde{X}_3(k)$ using Matlab:

```
% P5.5
%% P0505a.m
n1 = [0:39]; xtilde1 = [5*sin(0.1*pi*[0:19]),zeros(1,20)];
n3 = [0:79]; xtilde3 = [xtilde1, xtilde1]; N3 = length(n3);
[Xtilde3] = dft(xtilde3,N3); k3 = n3;
mag_Xtilde3 = abs(Xtilde3); pha_Xtilde3 =
angle(Xtilde3)*180/pi;
zei = find(mag_Xtilde3 < 0.00001);
pha_Xtilde3(zei) = zeros(1,length(zei));
Hf_1=figure('Units','normalized','position',[0.1,0.1,0.8,
0.8],'paperunits','inches');
%
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,6,5]);
set(Hf_1,'NumberTitle','off','Name','P5.5.1');
subplot(3,1,1);H_s1=stem(n3,xtilde3,'filled');
set(H_s1,'markersize',3);
title('One period of the periodic sequence
xtilde_3(n)','fontsize',10);
ylabel('Amplitude'); ntick = [n3(1):5:n3(N3),N3]';axis([-
1,N3,-6,6]);
set(gca,'XTickMode','manual','XTick',ntick,'fontsize',8)
subplot(3,1,2); H_s2 = stem(k3,mag_Xtilde3,'filled');
set(H_s2,'markersize',3);
axis([-1,N3,min(mag_Xtilde3),max(mag_Xtilde3)]);
title('Magnitude of Xtilde_3(k)','fontsize',10);
ylabel('Magnitude')
ktick=[k3(1):5:k3(N3),N3]';
set(gca,'XTickMode','manual','XTick',ktick)
subplot(3,1,3); H_s3 = stem(k3,pha_Xtilde3,'filled');
set(H_s3,'markersize',3);
title('Phase of Xtilde3(k)','fontsize',10); xlabel('k');
ylabel('Degrees');
ktick = [k3(1):5:k3(N3),N3]';axis([-1,N3,-180,180]);
set(gca,'XTickMode','manual','XTick',ktick)
```

```
set(gca,'YTickMode','manual','YTick',[-180;-90;0;90;180])
print -deps2 ../epsfiles/P0505a
```

Plots of $\tilde{x}_3(n)$ and $\tilde{X}_3(k)$ are shown in Figure 5.4.

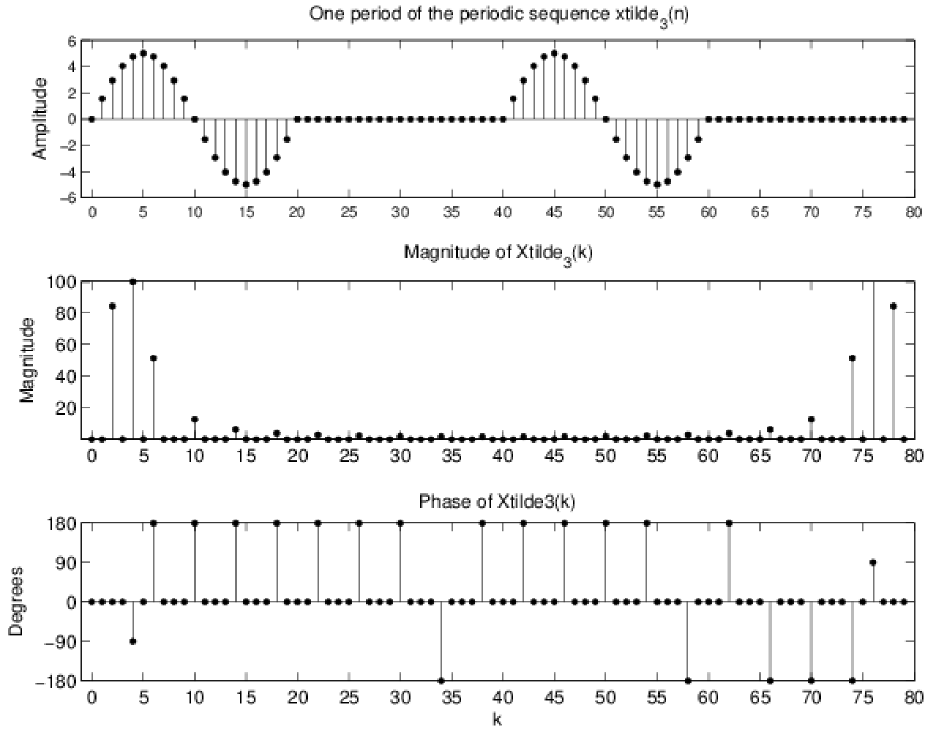


Figure 5.4: Plots of $\tilde{x}_3(n)$ and $\tilde{X}_3(k)$ in Problem 5.5a

2. Comparing the magnitude plot above with that of $\tilde{X}_1(k)$ in Problem (5), we observe that these plots are essentially similar. Plots of $\tilde{X}_3(k)$ have one zero between every sample of $\tilde{X}_1(k)$. (In general, for phase plots, we do get non-zero phase values when the magnitudes are zero. Clearly these phase values have no meaning and should be ignored. This happens because of a particular algorithm used by Matlab. We avoided this problem by using the **find** function.) This makes sense because sequences $\tilde{x}_1(n)$ and $\tilde{x}_3(n)$, when viewed over $-\infty < n < \infty$ interval, look exactly same. The effect of periodicity doubling is in the doubling of magnitude of each sample.

3. We can now generalize this argument. If

$$\tilde{x}_M(n) = \left\{ \underbrace{\tilde{x}_1(n), \tilde{x}_1(n), \ldots, \tilde{x}_1(n)}_{M \text{ times}} \right\}_{\text{PERIODIC}}$$

then there will be $(M - 1)$ zeros between samples of $\tilde{X}_M(k)$. The magnitudes of non-zero samples of $\tilde{X}_M(k)$ will be M times the magnitudes of the samples of $\tilde{X}_1(k)$, i.e.,

$$\begin{aligned}\tilde{X}_M(Mk) &= M\tilde{X}_1(k) &&, k = 0, 1, \ldots, N-1\\ \tilde{X}_M(k) &= 0 &&, k \neq 0, 1, \ldots, MN\end{aligned}$$

P5.6

Let $X(e^{jw})$ be the DTFT of a finite-length sequence

$$x(n) = \begin{cases} n+1, & 0 \le n \le 49;\\ 100-n, & 50 \le n \le 99;\\ 0, & \text{otherwise.}\end{cases}$$

1. Let

$$y_1(n) = \overset{\text{10-point}}{\text{IDFS}}\left[X(e^{j0}), X(e^{j2\pi/10}), X(e^{j4\pi/10}), \ldots, X(e^{j18\pi/10})\right]$$

Determine $y_1(n)$ using the frequency sampling theorem. Verify your answer using MATLAB.

2. Let

$$y_2(n) = \overset{\text{200-point}}{\text{IDFS}}\left[X(e^{j0}), X(e^{j2\pi/200}), X(e^{j4\pi/200}), \ldots, X(e^{j398\pi/200})\right]$$

Determine $y_2(n)$ using the frequency sampling theorem. Verify your answer using MATLAB.

3. Comment on your results in parts (a) and (b).

Solutions

1. Let

$$y_1(n) = \overset{\text{10-point}}{\text{IDFS}}\left[X(e^{j0}), X(e^{j2\pi/10}), X(e^{j4\pi/10}), \ldots, X(e^{j18\pi/10})\right]$$

which is a 10-point IDFS of ten samples of $X\left(e^{jw}\right)$ on the unit circle. Thus

$$\begin{aligned}y_1(n) = \sum_{r=-\infty}^{\infty} x(n-10r) = \{&1 + 11 + \cdots + 41 + 50 + 40 + \cdots + 10,\\ &2 + 12 + \cdots + 42 + 49 + \cdots + 9, \cdots\}_{\text{periodic}}\\ = \{&255, 255, \ldots, 255\}_{\text{periodic}}\end{aligned}$$

See the stem plot of $y_1(n)$ in Figure 5.5.

2. Let

$$y_2(n) = \overset{\text{200-point}}{\text{IDFS}}\left[X(e^{j0}), X(e^{j2\pi/200}), X(e^{j4\pi/200}), \ldots, X(e^{j398\pi/200})\right]$$

which is a 200-point IDFS of 200 samples of $X\left(e^{jw}\right)$ on the unit circle. Thus

$$y_2(n) = \left\{\begin{array}{ll} x(n), & 0 \le n \le 49;\\ 0, & 50 \le n \le 100.\end{array}\right\}_{\text{periodic}}$$

See the stem plot of $y_2(n)$ in Figure 5.5.

3. The sequence $y_1(n)$ is a 10-point aliasing version on $x(n)$ while $y_2(n)$ is a zero-padded version of $x(n)$.

```
% P5.6
%% P0506a.m
n = 0:99; x = [n(1:50)+1,100-n(51:100)];
N1 = 10; k1 = 0:N1-1; w1 = 2*pi*k1/N1;
Y1 = dtft(x,n,w1); y1 = real(idfs(Y1,N1));
mag_y1 = abs(y1);
n = 0:99; x = [n(1:50)+1,100-n(51:100)];
N2 = 200; k2 = 0:N2-1; w2 = 2*pi*k2/N2;
Y2 = dtft(x,n,w2); y2 = real(idfs(Y2,N2));
mag_y2 = abs(y2);
Hf_1                                                    =
figure('Units','normalized','position',[0.1,0.1,0.8,0.8],
...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.6.1');

subplot(3,1,1);H_s1                                     =
stem(n,x,'filled');set(H_s1,'markersize',3);
title('Original 100-point sequence x(n)','fontsize',10);
ylabel('Amptitude');ntick = [n(1):10:n(100),100]';axis([-
1,100,0,60]);
set(gca,'XTickMode','manual','XTick',ntick,'fontsize',8);

subplot(3,1,2);H_s2                                     =
stem(k1,mag_y1,'filled');set(H_s2,'markersize',3);
title('Aliased sequence y1(n)','fontsize',10);
ylabel('Amptitude');ktick  =  [k1(1):1:k1(10),10]';axis([-
1,11,0,max(mag_y1)+50]);
set(gca,'XTickMode','manual','XTick',ktick,'fontsize',8);

subplot(3,1,3);H_s3                                     =
stem(k2,mag_y2,'filled');set(H_s3,'markersize',3);
title('Unaliased sequence y2(n)','fontsize',10);
ylabel('Amptitude');mtick   =   [1,49,99,149,199]';axis([-
1,199,0,max(mag_y2)+10]);
set(gca,'XTickMode','manual','XTick',mtick,'fontsize',8);

print -deps2 ../epsfiles/P0506
```

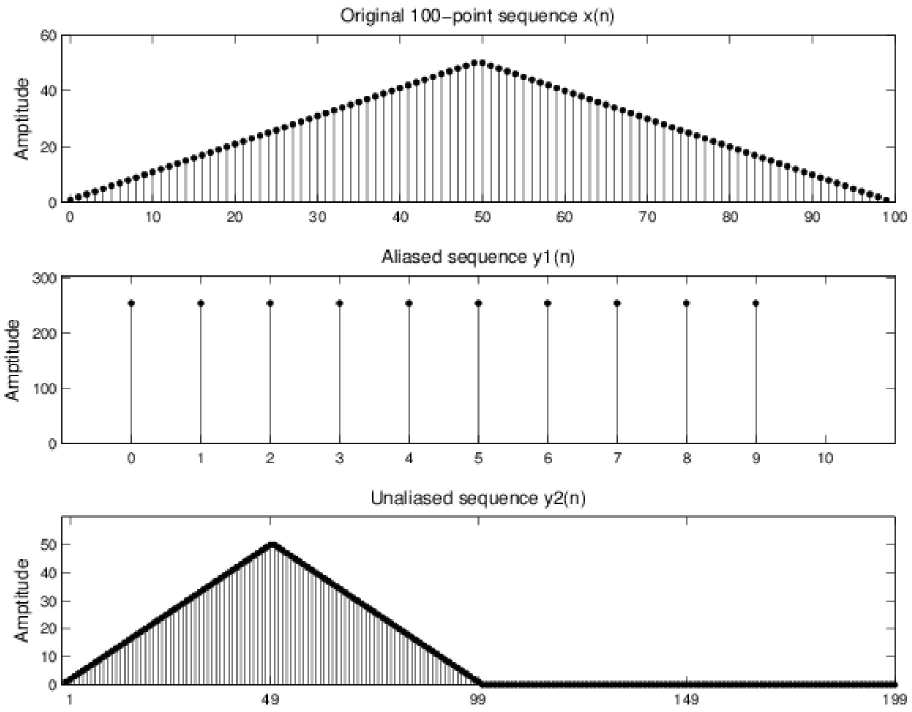


Figure 5.5: Plots of $y_1(n)$ and $y_2(n)$ in Problem 5.6

P5.7

Let $\tilde{x}(n)$ be a periodic sequence with period N and let

$$\tilde{y}(n) \triangleq \tilde{x}(-n) = \tilde{x}(N-n)$$

that is, $\tilde{y}(n)$ is a periodically folded version of $\tilde{x}(n)$. Let $\tilde{X}(k)$ and $\tilde{Y}(k)$ be the DFS sequences.

1. Show that

$$\tilde{Y}(k) = \tilde{X}(-k) = \tilde{X}(N-k)$$

that is, $\tilde{Y}(k)$ is also a periodically folded version of $\tilde{X}(k)$.

2. Let $\tilde{x}(n) = \{\underset{\uparrow}{2}, 4, 6, 1, 3, 5\}_{PERIODIC}$ with $N = 6$

(a) *Sketch* $\tilde{y}(n)$ for $0 \le n \le 5$.
(b) Compute $\tilde{X}(k)$ for $0 \le k \le 5$.
(c) Compute $\tilde{Y}(k)$ for $0 \le k \le 5$.
(d) Verify the relation in part 1.

Solutions

1. Consider

$$\tilde{Y}(k) = \text{DFS}\left[\tilde{y}(n)\right] = \sum_{n=0}^{N-1} \tilde{y}(n) W_N^{nk} = \sum_{n=0}^{N-1} \tilde{x}(-n) W_N^{nk} = \sum_{n=0}^{N-1} \tilde{x}(N-n) W_N^{nk}$$

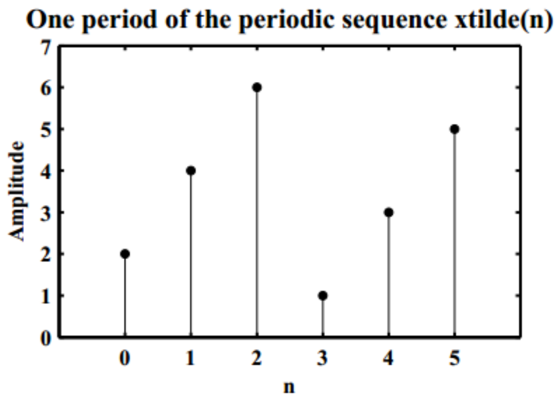
$$= \sum_{\ell=1}^{N} \tilde{x}(\ell) W_N^{(N-\ell)k} = \sum_{\ell=0}^{N-1} \tilde{x}(\ell) W_N^{Nk} W_N^{-\ell k} = \sum_{\ell=0}^{N-1} \tilde{x}(\ell) W_N^{-\ell k} \quad (\because \text{ periodic})$$

$$= \tilde{X}(-k) = \tilde{X}(N-k)$$

2. Let $\tilde{x}(n) = \{2, 4, 6, 1, 3, 5\}_{PERIODIC}$ with $N = 6$.

$\uparrow$

(a) Sketch of $\tilde{y}(n)$ for $0 \le n \le 5$:



(b) Computation of $\tilde{X}(k)$ for $0 \le k \le 5$:

```
% P5.7
x_tilde = [2,4,6,1,3,5];N = 6;
X_tilde = dft(x_tilde,N)
X_tilde =
  Columns 1 through 4
  21.0000 + 0.0000i   1.0000 - 1.7321i  -6.0000 + 3.4641i
1.0000 - 0.0000i
  Columns 5 through 6
  -6.0000 - 3.4641i   1.0000 + 1.7321i
```

(c) Computation of $\tilde{Y}(k)$ for $0 \le k \le 5$:

```
y_tilde = [x_tilde(1),fliplr(x_tilde(2:end))];
Y_tilde = dft(y_tilde,N)
Y_tilde =
  Columns 1 through 4
  21.0000 + 0.0000i   1.0000 + 1.7321i  -6.0000 - 3.4641i
1.0000 + 0.0000i
  Columns 5 through 6
  -6.0000 + 3.4641i   1.0000 - 1.7321i
```

(d) Matlab verification:

```
W_tilde = [X_tilde(1),fliplr(X_tilde(2:end))];
error = max(abs(Y_tilde-W_tilde))
error =
   2.4666e-14
```

P5.8

Consider the following finite-length sequence.

$$x(n) = \begin{cases} \text{sinc}^2\{(n-50)/2\}, & 0 \le n \le 100; \\ 0, & \text{else.} \end{cases}$$

1. Determine the DFT $X(k)$ of $x(n)$. Plot (using the **stem** function) its magnitude and phase.
2. Plot the magnitude and phase of the DTFT $X(e^{jw})$ of $x(n)$ using MATLAB.
3. Verify that the above DFT is the sampled version of $X(e^{jw})$. It might be helpful to combine the above two plots in one graph using the **hold** function.
4. Is it possible to reconstruct the DTFT $X(e^{jw})$ from the DFT $X(k)$? If possible, give the necessary interpolation formula for reconstruction. If not possible, state why this reconstruction cannot be done.

Solutions

1. DFT $X(k)$:

```
% P5.8
%% P0508a
clc;close all;
n = 0:100; xn = sinc((n-50)/2).^2; N = length(xn); % given
signal x(n)
Xk = dft(xn,N); k = 0:N-1; % DFT of x(n)
mag_Xk = abs(Xk); pha_Xk = angle(Xk)*180/pi; % Mag and
Phase of X(k)
zei = find(mag_Xk < 0.00001); % Set phase values to
pha_Xk(zei) = zeros(1,length(zei)); % zero when mag is zero
Hf_1                                                      =
figure('Units','normalized','position',[0.1,0.1,0.8,0.8],
...
 'paperunits','inches');
%
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,6,5]);
set(Hf_1,'NumberTitle','off','Name','P5.8');
subplot(2,1,1);     H_s1     =     stem(k,mag_Xk,'filled');
```

```
set(H_s1,'markersize',3);
set(gca,'XTick',[0:20:N],'fontsize',8); axis([0,N,0,2.5])
set(gca,'YTick',[0:0.5:2.5],'fontsize',8);
ylabel('Magnitude');
title('Magnitude plots of DFT and DTFT','fontsize',10);
hold on
subplot(2,1,2);      H_s2      =      stem(k,pha_Xk,'filled');
set(H_s2,'markersize',3);
set(gca,'XTick',[0:20:N],'fontsize',8);          axis([0,N,-
200,200])
set(gca,'YTick',[-180;-90;0;90;180],'fontsize',8);
xlabel('k'); ylabel('Degrees');
title('Phase plots of DFT and DTFT','fontsize',10); hold
on
```

The stem plot of $X(k)$ is shown in 5.6.

2. DTFT $X(e^{jw})$:

```
%% P0508b
[X,w] = freqz(xn,1,1000,'whole'); % DTFT of xn
mag_X = abs(X); pha_X = angle(X)*180/pi; % mag and phase
of DTFT
Dw = (2*pi)/N; % frequency resolution
subplot(2,1,1); plot(w/Dw,mag_X); grid
hold off
subplot(2,1,2); plot(w/Dw,pha_X); grid
hold off
print -deps2 ../epsfiles/P0508
```

The continuous plot of $X(e^{jw})$ is also shown in Figure 5.6.

3. Clearly, the DFT in part 1. is the sampled version of $X(e^{jw})$.

4. It is possible to reconstruct the DTFT from the DFT if length of the DFT is larger than or equal to the length of sequence $x(n)$. We can reconstruct using the complex interpolation formula

$$X\left(e^{j\omega}\right) = \sum_{k=0}^{N-1} X(k)\,\phi\left(\omega - \frac{2\pi k}{N}\right), \quad \text{where} \quad \phi(\omega) = e^{-j\omega(N-1)/2}\frac{\sin(\omega N/2)}{N\sin(\omega/2)}$$

For $N = 101$, we have

$$X\left(e^{j\omega}\right) = \sum_{k=0}^{100} X(k)\, e^{-j(50)\omega}\frac{\sin(50.5\omega)}{101\sin(\omega/2)}$$

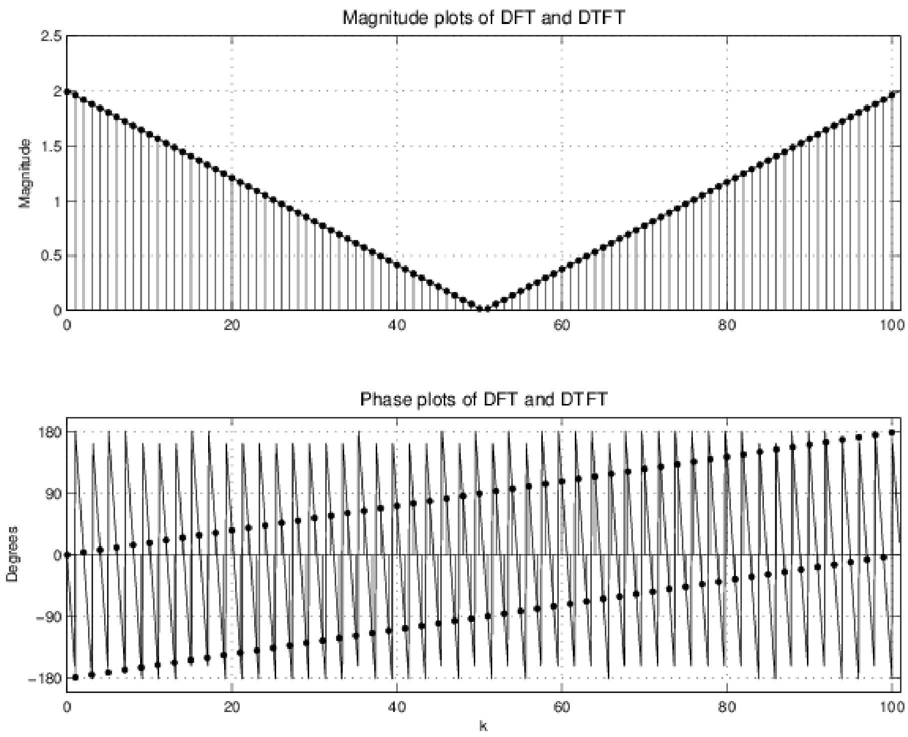


Figure 5.6: Plots of DTFT and DFT of signal in Problem 5.8

P5.9

Let a finite-length sequence be given by

$$x(n) = \begin{cases} 2e^{-0.9|n|}, & -5 \le n \le 5; \\ 0, & \text{otherwise.} \end{cases}$$

Plot the DTFT $X(e^{jw})$ of the above sequence using DFT as a computation tool. Choose the length N of the DFT so that this plot appears to be a smooth graph.

Solutions

```
% P5.9.m
clc; close all;
n = -5:5; xn = 2*exp(-0.9*abs(n)); N1 = length(xn);
N = 201; xn = [xn,zeros(1,N-N1)]; Xk = dft(xn,N); Xk =
real(Xk);
w = linspace(-pi,pi,N); Xk = fftshift(Xk);
Hf_1                                                    =
figure('Units','normalized','position',[0.1,0.1,0.8,0.8],
...
'paperunits','inches');
%
'color',[0,0,0],'paperunits','inches','paperposition',[0,
```

```
0,6,3]);
set(Hf_1,'NumberTitle','off','Name','P5.9');
plot(w/pi,Xk,'g','linewidth',1.5); axis([-1,1,-4,5]); hold
on;
plot([-1,1],[0,0],'w',[0,0],[-4,5],'w','linewidth',0.5);
title('DTFT   of   x(n)   =   2e^{-0.9|n|},   -5\leq   n\leq
5','fontsize',10);
xlabel('\omega/\pi','fontsize',10);
ylabel('Amplitude','fontsize',10);
print -deps2 ../epsfiles/P0509;
```

The plot of the DTFT $X(e^{jw})$ is shown in 5.7.

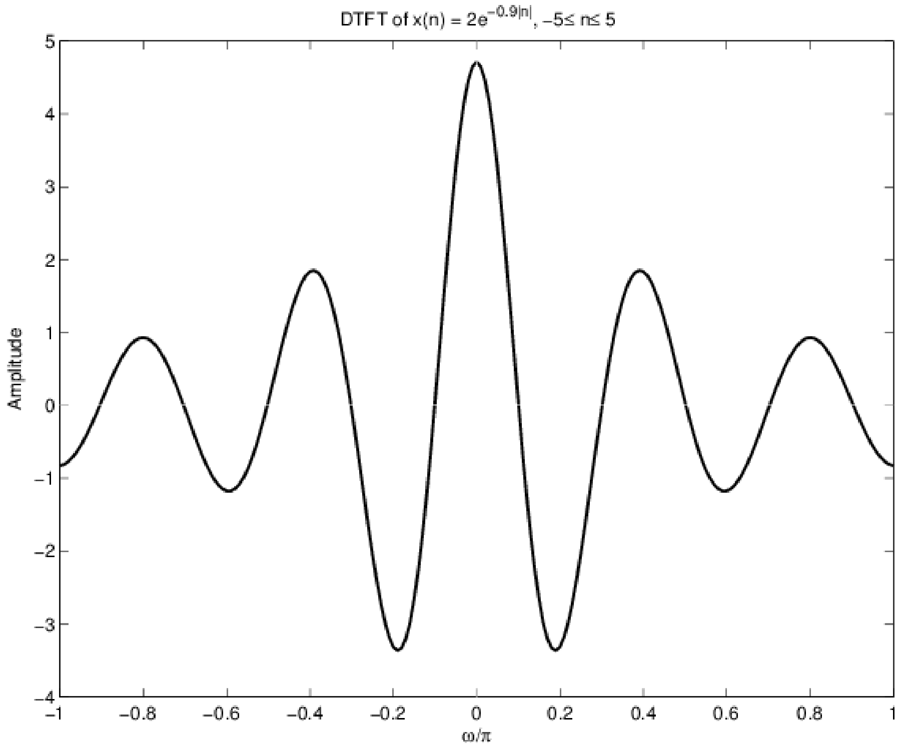


Figure 5.7: Plots of DTFT and DFT of signal in Problem 5.9

P5.10

Plot the DTFT magnitude and angle of each of the following sequences using the DFT as a computation tool. Make an educated guess about the length N so that your plots are meaningful.

1. $x(n) = (0.6)^{|n|}\,[u(n+10) - u(n-11)]$.
2. $x(n) = n(0.9)^n\,[u(n) - u(n-21)]$.
3. $x(n) = [\cos(0.5\pi n) + j\sin(0.5\pi n)][u(n) - u(n-51)]$.
4. $x(n) = \{1, 2, 3, \underset{\uparrow}{4}, 3, 2, 1\}$.
5. $x(n) = \{-1, -2, -3, \underset{\uparrow}{0}, 3, 2, 1\}$.

Solutions

1. $x_1(n) = (0.6)^{|n|}[u(n+10) - u(n-10)]$. Matlab script:

```
% P5.10
%% P0510a.m
n1 = [-10:10]; x1 = (0.6).^abs(n1); N1 = length(n1); N =
200; % Length of DFT
x1 = [x1(11:end), zeros(1,N-N1), x1(1:10)]; % Assemble x1
[X1] = fft(x1,N); w = (0:N/2)*2*pi/N;
mag_X1        =        abs(X1(1:N/2+1));        pha_X1        =
angle(X1(1:N/2+1))*180/pi;
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
%'color',[0,0,0],'paperunits','inches','paperposition',[0
,0,6,4]);
set(Hf_1,'NumberTitle','off','Name','P5.10.1');
subplot(2,1,1);       plot(w/pi,mag_X1,'g','linewidth',1);
axis([0,1,0,11]);
title('Magnitude      of      DTFT      X_1(e^{j\omega})');
ylabel('Magnitude');
subplot(2,1,2);       plot(w/pi,pha_X1,'g','linewidth',1);
axis([0,1,-200,200]);
title('Angle of DTFT X_1(e^{j\omega})'); ylabel('Degrees');
xlabel('\omega/\pi');
print -deps2 ../EPSFILES/P0510a
```

The plot of the DTFT $X_1(e^{jw})$ is shown in 5.8.

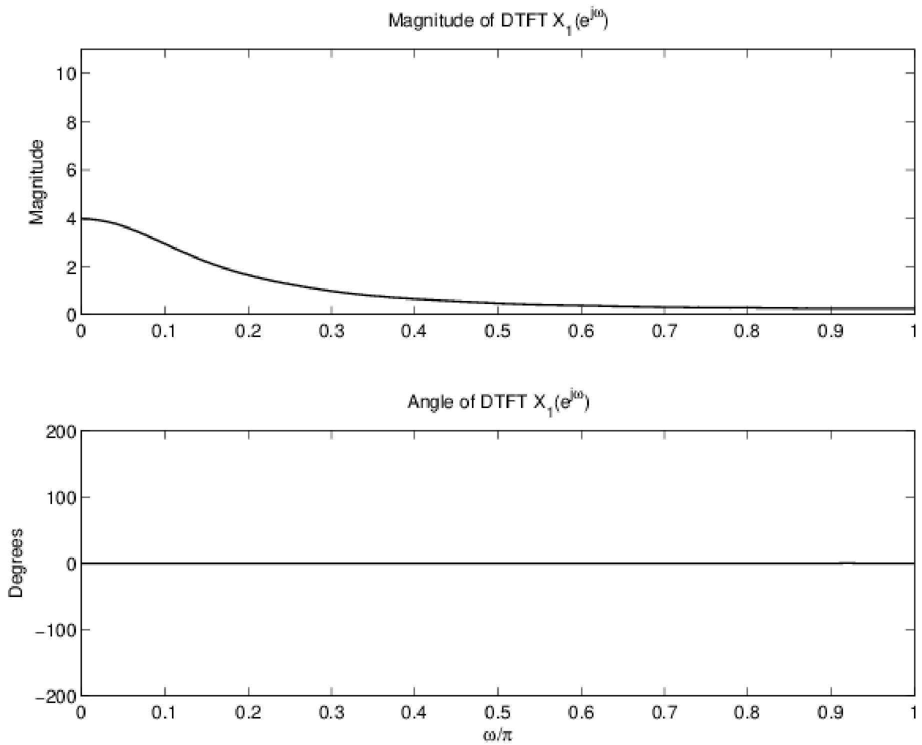


Figure 5.8: Plots of DTFT magnitude and phase in Problem 5.10.1

2. $x_2(n) = n(0.9)^n$, $0 \le n \le 20$. Matlab script:

```
%% P0510b.m
n2 = [0:20]; x2 = n2.*(0.9).^n2; N2 = length(n2); N = 400; % Length of DFT
x2 = [x2,zeros(1,N-N2)]; % Assemble x2
[X2] = fft(x2,N); w = (0:N/2)*2*pi/N;
mag_X2 = abs(X2(1:N/2+1)); pha_X2 = angle(X2(1:N/2+1))*180/pi;
Hf_2 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_2,'NumberTitle','off','Name','P5.10.2');
subplot(2,1,1);
plot(w/pi,mag_X2,'g','linewidth',1); %axis([0,1,0,5]);
title('Magnitude of DTFT X_2(e^{j\omega})');
ylabel('Magnitude');
subplot(2,1,2); plot(w/pi,pha_X2,'g','linewidth',1);
axis([0,1,-200,200]);
title('Angle of DTFT X_2(e^{j\omega})'); ylabel('Degrees');
xlabel('\omega/\pi');
print -deps2 ../EPSFILES/P0510b
```

The plot of the DTFT $X_2(e^{jw})$ is shown in 5.9.

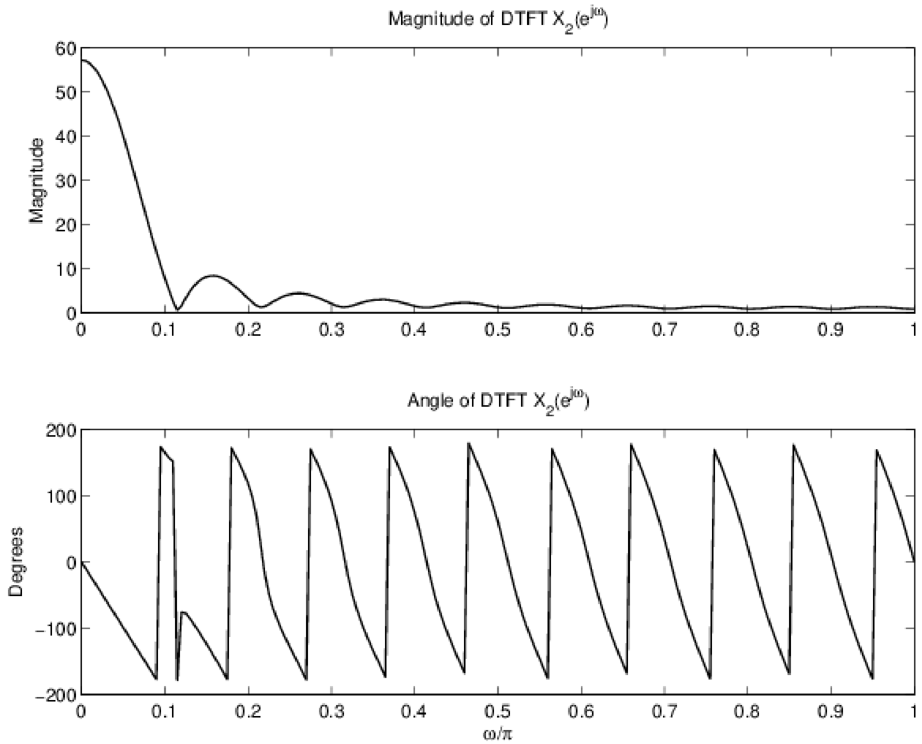


Figure 5.9: Plots of DTFT magnitude and phase in Problem 5.10.2

3. $x_3(n) = \cos(0.5\pi n) + j\sin(0.5\pi n)$, $0 \le n \le 50$. Matlab script:

```
%% P0510c.m
n3 = [0:50]; x3 = cos(0.5*pi*n3)+j*sin(0.5*pi*n3); N = 500;%
Length of DFT
N3 = length(n3); x3 = [x3,zeros(1,N-N3)]; % Assemble x3
% [X3] = fft(x3,N); w = (0:N/2)*2*pi/N; the phase is
different between fft and dft
[X3] = dft(x3,N); w = (0:N/2)*2*pi/N;
mag_X3 = abs(X3(1:N/2+1)); pha_X3 =
angle(X3(1:N/2+1))*180/pi;
Hf_3 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_3,'NumberTitle','off','Name','P5.10.3');
subplot(2,1,1);
plot(w/pi,mag_X3,'g','linewidth',1); %axis([0,1,0,7000]);
title('Magnitude of DTFT X_3(e^{j\omega})');
ylabel('Magnitude');
subplot(2,1,2); plot(w/pi,pha_X3,'g','linewidth',1);
axis([0,1,-200,200]);
title('Angle of DTFT X_3(e^{j\omega})'); ylabel('Degrees');
xlabel('\omega/\pi');
print -deps2 ../EPSFILES/P0510c
```

The plot of the DTFT $X_3(e^{jw})$ is shown in 5.10.

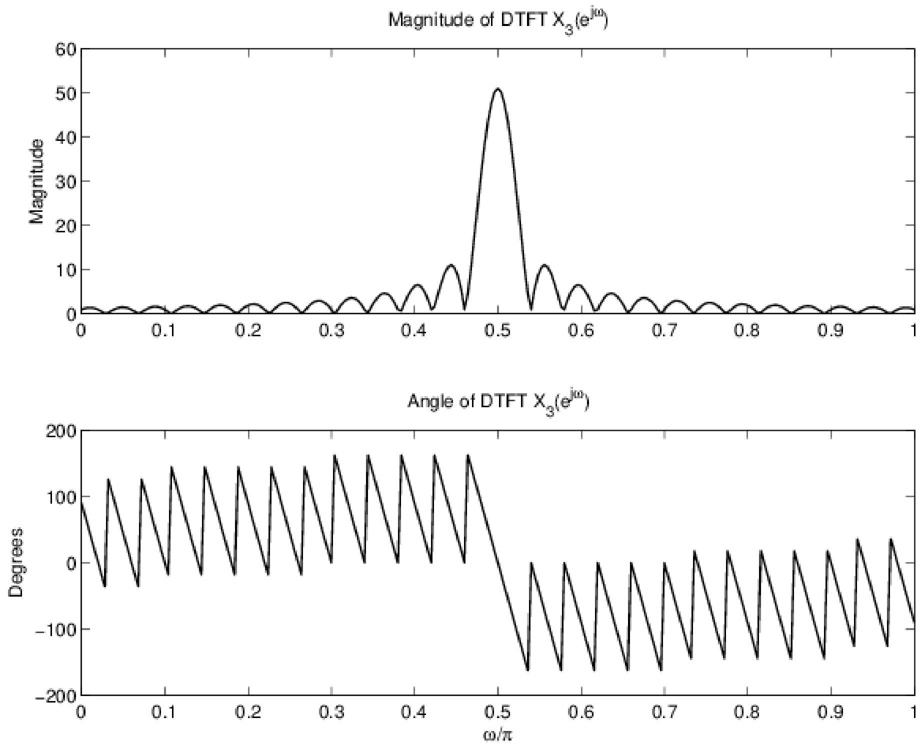


Figure 5.10: Plots of DTFT magnitude and phase in Problem 5.10.3

4. $x_4(n) = \{1, 2, 3, 4, 3, 2, 1\}$. Matlab script:

$\uparrow$ (under 4)

```
%% P0510d.m
n4 = [-3:3]; x4 = [1,2,3,4,3,2,1]; N4 = length(n4); N =
100; % Length of DFT
% [X4] = fft([x4, zeros(1,N-N4)],N); w = (0:N/2)*2*pi/N;
[X4] = dft([x4, zeros(1,N-N4)],N); w = (0:N/2)*2*pi/N;
mag_X4 = abs(X4(1:N/2+1)); pha_X4 =
angle(X4(1:N/2+1))*180/pi;
Hf_4 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_4,'NumberTitle','off','Name','P5.10.4');
subplot(2,1,1); plot(w/pi,mag_X4,'g','linewidth',1);
axis([0,1,0,20]);
title('Magnitude of DTFT X_4(e^{j\omega})');
ylabel('Magnitude');
subplot(2,1,2); plot(w/pi,pha_X4,'g','linewidth',1);
axis([0,1,-200,200]);
title('Angle of DTFT X_4(e^{j\omega})'); ylabel('Degrees');
xlabel('\omega/\pi');
print -deps2 ../EPSFILES/P0510d
```

The plot of the DTFT $X_4(e^{jw})$ is shown in 5.11.

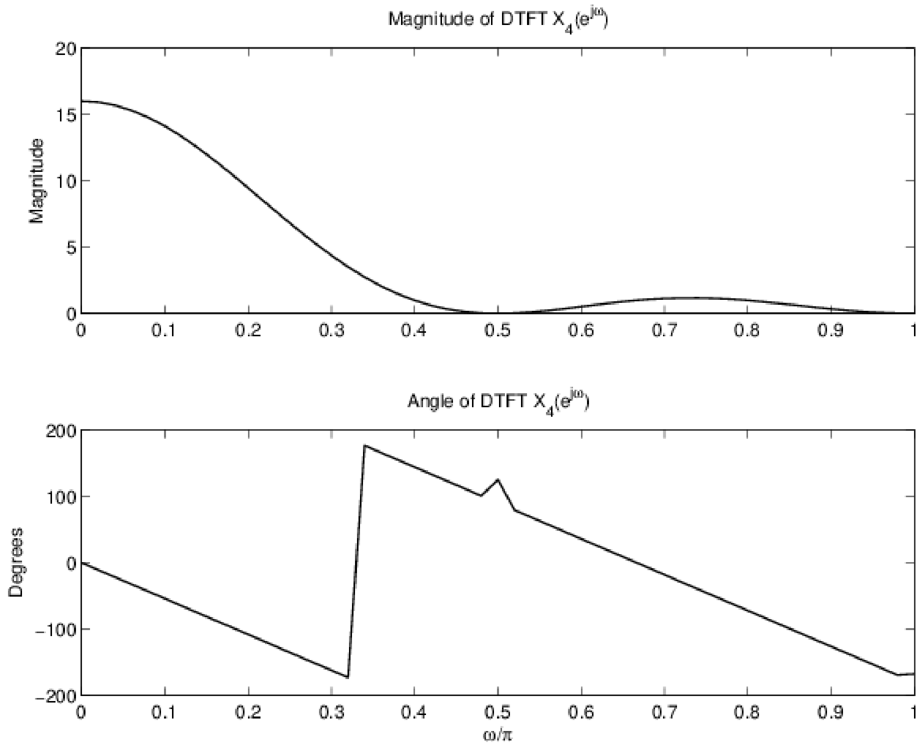


Figure 5.11: Plots of DTFT magnitude and phase in Problem 5.10.4

5. $x_5(n) = \{-1, -2, -3, 0, 3, 2, 1\}$. Matlab script:
$\uparrow$

```
%% P0510e.m
n5 = [-3:3]; x5 = [-1,-2,-3,0,3,2,1]; N5 = length(n5); N =
100; % Length of DFT
% [X5] = fft([x5, zeros(1,N-N5)],N); w = (0:N/2)*2*pi/N;
the phase is different between fft and dft
[X5] = dft([x5, zeros(1,N-N5)],N); w = (0:N/2)*2*pi/N;
mag_X5 = abs(X5(1:N/2+1)); pha_X5 =
angle(X5(1:N/2+1))*180/pi;
Hf_5 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_5,'NumberTitle','off','Name','P5.10.5');
subplot(2,1,1); plot(w/pi,mag_X5,'g','linewidth',1);
axis([0,1,0,20]);
title('Magnitude of DTFT X_5(e^{j\omega})');
ylabel('Magnitude');
subplot(2,1,2); plot(w/pi,pha_X5,'g','linewidth',1);
axis([0,1,-200,200]);
title('Angle of DTFT X_5(e^{j\omega})'); ylabel('Degrees');
xlabel('\omega/\pi');
print -deps2 ../EPSFILES/P0510e
```

The plot of the DTFT $X_5(e^{jw})$ is shown in 5.12.

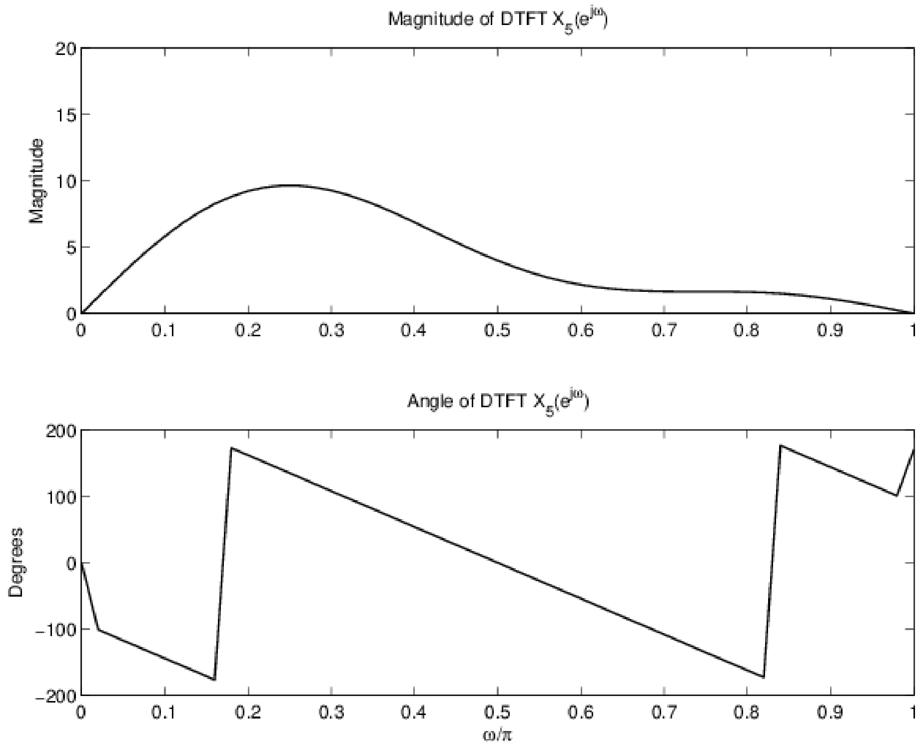


Figure 5.12: Plots of DTFT magnitude and phase in Problem 5.10.5

P5.11

Let $H(e^{jw})$ be the frequency response of a real, causal discrete-time LSI system.

1. If

$$\operatorname{Re}\left\{H\left(e^{j\omega}\right)\right\} = \sum_{k=0}^{5} (0.9)^k \cos(k\omega)$$

determine the impulse response $h(n)$ analytically. Verify your answer using DFT as a computation tool. Choose the length N appropriately.

2. If

$$\operatorname{Im}\left\{H\left(e^{j\omega}\right)\right\} = \sum_{\ell=0}^{5} 2\ell \sin(\ell\omega), \quad \text{and} \quad \int_{-\pi}^{\pi} H(e^{j\omega})d\omega = 0$$

determine the impulse response $h(n)$ analytically. Verify your answer using DFT as a computation tool. Again choose the length N appropriately.

Solutions

1. It is known that $\operatorname{Re}\left\{H(e^{jw})\right\} = \sum_{k=0}^{5} (0.9)^k \cos wk$ .Consider

$$\text{Re}\left\{H\left(e^{j\omega}\right)\right\} = \text{Re}\left\{\sum_{k=0}^{\infty} h(k)e^{-j\omega k}\right\} = \sum_{k=0}^{\infty} h(k)\,\text{Re}\left\{e^{-j\omega k}\right\} = \sum_{k=0}^{\infty} h(k)\cos\omega k$$

Comparing with the given expression, we obtain

$$h(n) = \begin{cases} (0.9)^n, & 0 \le n \le 5 \\ 0, & \text{else} \end{cases}$$

Matlab verification:

```
% P5.11
%% P0511a.m
n = 0:5; h = (0.9).^n; N1 = length(h); N = 100; h =
[h,zeros(1,N-N1)];
H = dft(h,N); Hr = real(H);
k = [0:5]; w = linspace(0,2*pi,N+1);
Hr_check = (0.9.^k)*cos(k'*w(1:end-1));
error = max(abs(Hr-Hr_check))
 error =
   3.9968e-14
```

2. It is known that $\text{Im}\left\{H(e^{jw})\right\} = \sum_{l=0}^{5} 2l \sin wl$ and $\int_{-\pi}^{\pi} H(e^{jw})dw = 0$. From the second condition

$$\int_{-\pi}^{\pi} H\left(e^{j\omega}\right) d\omega = h\left(0\right) = 0$$

Consider

$$\text{Im}\left\{H\left(e^{j\omega}\right)\right\} = \text{Im}\left\{\sum_{\ell=0}^{\infty} h(\ell)e^{-j\omega\ell}\right\} = \sum_{\ell=0}^{\infty} h\left(\ell\right)\text{Im}\left\{e^{-j\omega\ell}\right\} = -\sum_{\ell=0}^{\infty} h\left(\ell\right)\sin\omega\ell$$

Comparing with the given expression, we obtain

$$h(n) = \begin{cases} -2n, & 0 \le n \le 5 \\ 0, & \text{else} \end{cases}$$

Matlab verification:

```
%% P0511b.m
n = 0:5; h = -2*n; N1 = length(h); N = 100; h = [h,zeros(1,N-
N1)];
H = dft(h,N); Hi = imag(H);
l = [0:5]; w = linspace(0,2*pi,N+1);
Hi_check = 2*l*sin(l'*w(1:end-1));
error = max(abs(Hi-Hi_check))
error =
   3.8014e-13
```

P5.12

Let $X(k)$ denote the N-point DFT of an N-point sequence $x(n)$. The DFT $X(k)$ itself is an N-point sequence.

1. If the DFT of $X(k)$ is computed to obtain another N-point sequence $x_1(n)$, show that

$$x_1(n) = Nx((-n))_N, \quad 0 \le n \le N-1$$

2. Using this property, design a MATLAB function to implement an N-point circular folding operation $x_2(n) = x_1((-n))N$. The format should be

```
x2 = circfold(x1,N)
% Circular folding using DFT
% x2 = circfold(x1,N)
% x2 = circularly folded output sequence
% x1 = input sequence of length <= N
% N = circular buffer length
```

3. Determine the circular folding of the following sequence.

$$x_1(n) = \{1, 3, 5, 7, 9, -7, -5, -3, -1\}$$

Solutions

1. The N-point DFT of $x(n)$: $X(k) = \sum_{m=0}^{N-1} x(m) W_N^{mk}$ .The N-point DFT of $X(k)$: $y(n) = \sum_{k=0}^{N-1} X(k) W_N^{kn}$ .Hence,

$$\begin{aligned}
y(n) &= \sum_{k=0}^{N-1} \left\{ \sum_{m=0}^{N-1} x(m) W_N^{mk} \right\} W_N^{kn} = \sum_{m=0}^{N-1} x(m) \sum_{k=0}^{N-1} W_N^{mk} W_N^{kn}, \quad 0 \le n \le N-1 \\
&= \sum_{m=0}^{N-1} x(m) \sum_{k=0}^{N-1} W_N^{(m+n)k} = \sum_{m=0}^{N-1} x(m) \sum_{r=-\infty}^{\infty} N\delta(m+n-rN), 0 \le n \le N-1 \\
&= N \sum_{r=-\infty}^{\infty} x(-n+rN) = Nx((-n))_N, 0 \le n \le N-1
\end{aligned}$$

This means that $y(n)$ is a "circularly folded and amplified (by N)" version of $x(n)$. Continuing further, if we take two more DFTs of $x(n)$ then

$$x(n) \longrightarrow \boxed{\begin{matrix} N\text{-point} \\ \text{DFT} \end{matrix}} \longrightarrow \boxed{\begin{matrix} N\text{-point} \\ \text{DFT} \end{matrix}} \longrightarrow \boxed{\begin{matrix} N\text{-point} \\ \text{DFT} \end{matrix}} \longrightarrow \boxed{\begin{matrix} N\text{-point} \\ \text{DFT} \end{matrix}} \longrightarrow N^2 x(n)$$

Therefore, if a given DFT function is working correctly then four successive applications of this function on any arbitrary signal will produce the same signal (multiplied by N^2). This approach can be used to verify a DFT function.

2. Matlab function for circular folding:

```
function x2 = circfold(x1,N)
% Circular folding using DFT
% x2 = circfold(x1,N)
% x2 = circularly folded output sequence
% x1 = input sequence of length <= N
% N = circular buffer length
if any(imag(x1) ~=0)
x2 = dft(dft(x1,N),N)/N;
else
    x2 = real(dft(dft(x1,N),N))/N;
end

% function x2 = circfold(x1,N)
% % Circular folding without using DFT
% % x2 = circfold(x1,N)
% % x2 = circularly folded output sequence
% % x1 = input sequence of length <= N
% % N = circular buffer length
% n = 0:N-1;
% % x2 = zeros(1,length(x1));
% x2 = x1(mod(-n,length(x1))+1);
```

3. Matlab verification:

```
x =
     1     3     5     7     9    -7    -5    -3    -1
Y =
  Columns 1 through 8
    1.0000   -1.0000   -3.0000   -5.0000   -7.0000    9.0000
7.0000    5.0000
  Column 9
    3.0000
```

P5.13

Let $X(k)$ be an N-point DFT of an N-point sequence $x(n)$. Let N be an even integer.

1. If $x(n) = x(n + N/2)$ for all n, then show that $X(k) = 0$ for k odd (i.e., nonzero for k even). Verify this result for $x(n) = \{1,2, -3, 4, 5, 1, 2, -3, 4, 5\}$.

2. If $x(n) = -x(n + N/2)$ for all n, then show that $X(k) = 0$ for k even (i.e., nonzero for k odd). Verify this result for $x(n) = \{1,2, -3, 4, 5, -1, -2, 3, -4, -5\}$.

Solutions

1. Given that $x(n) = x(n + N/2)$ for all n, consider

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk} \\
&= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{(n+N/2)k} \quad [\because n \to n+N/2] \\
&= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=0}^{N/2-1} x(n) W_N^{nk} W_N^{Nk/2} \quad [\because x(n) = x(n+N/2)] \\
&= \sum_{n=0}^{N/2-1} x(n) \left\{1 + (-1)^k\right\} W_N^{nk} \quad [\because W_N^{N/2} = -1] = \begin{cases} 0, & k \text{ odd;} \\ \text{Non-zero}, & k \text{ even.} \end{cases}
\end{aligned}$$

Verification using $x(n)$ = {1, 2, -3, 4, 5, 1, 2, -3, 4, 5}:

```
% P5.13
%% 0513a
% Verification using x(n) = {1, 2, -3, 4, 5, 1, 2, -3, 4,
5}:
x = [1,2,-3,4,5,1,2,-3,4,5]; N = length(x); X = dft(x,N)
X =
  Columns 1 through 4
  18.0000 + 0.0000i  -0.0000 + 0.0000i   4.7082 +13.9353i
-0.0000 + 0.0000i
  Columns 5 through 8
  -8.7082 - 9.7881i   0.0000 - 0.0000i  -8.7082 + 9.7881i
-0.0000 - 0.0000i
  Columns 9 through 10
   4.7082 -13.9353i   0.0000 - 0.0000i
```

2. Given that $x(n) = -x(n + N/2)$ for all n, consider

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=N/2}^{N-1} x(n) W_N^{nk} \\
&= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} + \sum_{n=0}^{N/2-1} x(n+N/2) W_N^{(n+N/2)k} \quad [\because n \to n+N/2] \\
&= \sum_{n=0}^{N/2-1} x(n) W_N^{nk} - \sum_{n=0}^{N/2-1} x(n) W_N^{nk} W_N^{Nk/2} \quad [\because x(n) = -x(n+N/2)] \\
&= \sum_{n=0}^{N/2-1} x(n) \left\{1 - (-1)^k\right\} W_N^{nk} \quad [\because W_N^{N/2} = -1] \\
&= \begin{cases} 0, & k \text{ even;} \\ \text{Non-zero}, & k \text{ odd.} \end{cases}
\end{aligned}$$

Verification using $x(n)$ = {1, 2, -3, 4, 5, -1, -2, 3, -4, -5}.

```
%% 0513b
% Verification using x(n) = {1, 2, -3, 4, 5, -1, -2, 3, -
```

```
4, -5}.
x = [1,2,-3,4,5,-1,-2,3,-4,-5]; N = length(x); X =
dft(x,N)
X =
  Columns 1 through 4
   0.0000 + 0.0000i  -7.1803 -10.1311i  -0.0000 - 0.0000i
15.1803 -12.1392i
  Columns 5 through 8
   0.0000 + 0.0000i  -6.0000 - 0.0000i   0.0000 - 0.0000i
15.1803 +12.1392i
  Columns 9 through 10
  -0.0000 + 0.0000i  -7.1803 +10.1311i
```

P5.14

Let $X(k)$ be an N-point DFT of an N-point sequence $x(n)$. Let $N = 4\nu$ where ν is an integer.
1. If $x(n) = x(n + \nu)$ for all n, then show that $X(k)$ is nonzero for $k = 4$ for $0 \leq \quad \leq \nu - 1$. Verify this result for $x(n) = \{1,2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3\}$.
2. If $x(n) = -x(n + \nu)$ for all n, then show that $X(k)$ is nonzero for $k = 4 + 2$ for $0 \leq \quad \leq \nu - 1$. Verify this result for $x(n) = \{1, 2, 3, -1, -2, -3, 1, 2, 3, -1, -2, -3\}$.

Solutions

1. It is given that $x(n) = x(n + \nu)$ for all n. Let $n = m + p\nu$; $0 \leq m \leq \nu - 1$, $0 \leq p \leq 3$, then

$$x(n) = x(n+\nu) \Rightarrow x(m+p\nu) = x(m), \quad 0 \leq m \leq \nu - 1 \tag{5.1}$$

Now the DFT $X(k)$ can be written as

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{p=0}^{3}\sum_{m=0}^{\nu-1} x(m+p\nu) W_N^{(m+p\nu)k} = \sum_{p=0}^{3}\sum_{m=0}^{\nu-1} x(m)\, W_N^{mk}\, W_N^{p\nu k} \qquad [\because (5.1)] \\
&= \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \sum_{p=0}^{3} W_N^{p\nu k} = \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \sum_{p=0}^{3} \left(W_N^{\nu k}\right)^p = \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \left[\frac{1 - W_N^{Nk}}{1 - W_N^{\nu k}}\right] \\
&= \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \left[\frac{1 - W_N^{Nk}}{1 - W_N^{N(k/4\ell)}}\right] = \begin{cases} \text{Non-zero}, & k = 4\ell \text{ for } 0 \leq \ell \leq \nu - 1; \\ 0, & k \neq 4\ell \text{ for } 0 \leq \ell \leq \nu - 1. \end{cases}
\end{aligned}$$

Verification for $x(n) = \{1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3\}$.

```
% P5.14
%% P0514a.m
% Verification for x(n) = {1, 2, 3, 1, 2, 3, 1, 2, 3, 1,
2, 3}.
x = [1,2,3,1,2,3,1,2,3,1,2,3]; N = length(x); X = dft(x,N)
```

```
X =
  Columns 1 through 4
  24.0000 + 0.0000i  -0.0000 - 0.0000i  -0.0000 + 0.0000i
-0.0000 + 0.0000i
  Columns 5 through 8
  -6.0000 + 3.4641i   0.0000 - 0.0000i  -0.0000 - 0.0000i
-0.0000 - 0.0000i
  Columns 9 through 12
  -6.0000 - 3.4641i   0.0000 - 0.0000i   0.0000 - 0.0000i
0.0000 - 0.0000i
```

2. It is given that $x(n) = -x(n + \nu)$ for all n. Let $n = m + p\nu$; $0 \leq m \leq \nu - 1$, $0 \leq p \leq 3$, then

$$x(n) = -x(n+\nu) \Rightarrow x(m+p\nu) = (-1)^p x(m), \quad 0 \leq m \leq \nu - 1,\ 0 \leq p \leq 3 \tag{5.2}$$

Now the DFT $X(k)$ can be written as

$$\begin{aligned} X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{p=0}^{3}\sum_{m=0}^{\nu-1} x(m+p\nu) W_N^{(m+p\nu)k} = \sum_{p=0}^{3}\sum_{m=0}^{\nu-1} (-1)^p x(m)\, W_N^{mk}\, W_N^{p\nu k} \qquad [\because (5.2)] \\ &= \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \sum_{p=0}^{3} (-1)^p W_N^{p\nu k} = \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \sum_{p=0}^{3} \left(W_N^{-N/2} W_N^{\nu k}\right)^p \qquad [\because -1 = W_N^{-N/2}] \\ &= \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \left[\frac{1 - W_N^{N(k-2)}}{1 - W_N^{\nu(k-2)}}\right] = \begin{cases} \text{Non-zero}, & k = 4\ell + 2 \text{ for } 0 \leq \ell \leq \nu - 1; \\ 0, & k \neq 4\ell + 2 \text{ for } 0 \leq \ell \leq \nu - 1. \end{cases} \end{aligned}$$

Verification for $x(n) = \{1, 2, 3, -1, -2, -3, 1, 2, 3, -1, -2, -3\}$.

```
%% P0514b.m
% Verification for x(n) = {1, 2, 3, -1, -2, -3, 1, 2, 3, -
1, -2, -3}.
x = [1,2,3,-1,-2,-3,1,2,3,-1,-2,-3]; N = length(x); X =
dft(x,N)
X =
  Columns 1 through 4
   0.0000 + 0.0000i   0.0000 - 0.0000i   2.0000 -17.3205i
0.0000 + 0.0000i
  Columns 5 through 8
   0.0000 + 0.0000i   0.0000 + 0.0000i   8.0000 + 0.0000i
-0.0000 + 0.0000i
  Columns 9 through 12
  -0.0000 + 0.0000i   0.0000 + 0.0000i   2.0000 +17.3205i
-0.0000 - 0.0000i
```

P5.15

Let $X(k)$ be an N-point DFT of an N-point sequence $x(n)$. Let $N = 2\mu\nu$ where μ and ν are integers.
1. If $x(n) = x(n + \nu)$ for all n, then show that $X(k)$ is nonzero for $k = 2(\mu l)$ for $0 \leq l \leq \nu - 1$. Verify this result for $x(n) = \{1, -2, 3, 1, -2, 3, 1, -2, 3, 1, -2, 3, 1, -2, 3, 1, -2, 3\}$.

2. If $x(n) = -x(n + \nu)$ for all n, then show that $X(k)$ is nonzero for $k = 2(\mu l + 1)$ for $0 \le l \le \nu - 1$. Verify this result for $x(n) = \{1, -2, 3, -1, 2, -3, 1, -2, 3, -1, 2, -3, 1, -2, 3, -1, 2, -3\}$

Solutions

Let $X(k)$ be an N-point DFT of an N-point sequence $x(n)$. Let $N = 2\mu\nu$ where μ and ν are integers.
1. It is given that $x(n) = x(n + \nu)$ for all n. Let $n = m + p\nu$; $0 \le m \le \nu - 1$, $0 \le p \le (2\mu - 1)$, then

$$x(n) = x(n+\nu) \Rightarrow x(m+p\nu) = x(m), \quad 0 \le m \le \nu - 1,\ 0 \le p \le (2\mu - 1) \tag{5.3}$$

Now the DFT $X(k)$ can be written as

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{p=0}^{2\mu-1} \sum_{m=0}^{\nu-1} x(m+p\nu) W_N^{(m+p\nu)k} = \sum_{p=0}^{2\mu-1} \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk}\, W_N^{p\nu k} \qquad [\because (5.3)] \\
&= \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \sum_{p=0}^{2\mu-1} W_N^{p\nu k} = \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \sum_{p=0}^{2\mu-1} \left(W_N^{\nu k}\right)^p = \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \left[\frac{1 - W_N^{Nk}}{1 - W_N^{\nu k}}\right] \\
&= \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \left[\frac{1 - W_N^{Nk}}{1 - W_N^{N(k/2\mu)}}\right] = \begin{cases} \text{Non-zero}, & k = (2\mu)\ell \text{ for } 0 \le \ell \le \nu - 1; \\ 0, & k \ne (2\mu)\ell \text{ for } 0 \le \ell \le \nu - 1. \end{cases}
\end{aligned}$$

Verification for $x(n) = \{1, -2, 3, 1, -2, 3, 1, -2, 3, 1, -2, 3, 1, -2, 3, 1, -2, 3\}$.

```
% P5.15
%% 0515a.m
% Verification for x(n) = {1, -2, 3, 1, -2, 3, 1, -2, 3,
1, -2, 3, 1, -2, 3, 1, -2, 3}.
x = [1,-2,3,1,-2,3,1,-2,3,1,-2,3,1,-2,3,1,-2,3]; N =
length(x); X = dft(x,N)
```

2. It is given that $x(n) = -x(n + \nu)$ for all n. Let $n = m + p\nu$; $0 \le m \le \nu - 1$, $0 \le p \le (2\mu - 1)$, then

$$x(n) = -x(n+\nu) \Rightarrow x(m+p\nu) = (-1)^p x(m), \quad 0 \le m \le \nu - 1,\ 0 \le p \le (2\mu - 1) \tag{5.4}$$

Now the DFT $X(k)$ can be written as

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n) W_N^{nk} = \sum_{p=0}^{2\mu-1} \sum_{m=0}^{\nu-1} x(m+p\nu) W_N^{(m+p\nu)k} = \sum_{p=0}^{2\mu-1} \sum_{m=0}^{\nu-1} (-1)^p x(m)\, W_N^{mk}\, W_N^{p\nu k} \qquad [\because (5.4)] \\
&= \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \sum_{p=0}^{2\mu-1} (-1)^p\, W_N^{p\nu k} = \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \sum_{p=0}^{2\mu-1} \left(W_N^{-N/2} W_N^{\nu k}\right)^p \qquad [\because -1 = W_N^{-N/2}] \\
&= \sum_{m=0}^{\nu-1} x(m)\, W_N^{mk} \left[\frac{1 - W_N^{N(k-2)}}{1 - W_N^{N(k-2)/(2\mu)}}\right] = \begin{cases} \ne 0, & k = 2\mu\ell + 2 = 2(\mu\ell + 1),\ 0 \le \ell \le \nu - 1; \\ 0, & k \ne 2\mu\ell + 2 = 2(\mu\ell + 1),\ 0 \le \ell \le \nu - 1. \end{cases}
\end{aligned}$$

Verification for $x(n) = \{1, -2, 3, -1, 2, -3, 1, -2, 3, -1, 2, -3, 1, -2, 3, -1, 2, -3\}$.

```
%% 0515b.m
% Verification for x(n) = {1, -2, 3, -1, 2, -3, 1, -2, 3,
-1, 2, -3, 1, -2, 3, -1, 2, -3}.
x = [1,-2,3,-1,2,-3,1,-2,3,-1,2,-3,1,-2,3,-1,2,-3]; N =
length(x); X = dft(x,N)
```

```
X =
  Columns 1 through 4
   0.0000 + 0.0000i  -0.0000 + 0.0000i  -0.0000 + 0.0000i
-9.0000 - 5.1962i
  Columns 5 through 8
   0.0000 - 0.0000i   0.0000 - 0.0000i   0.0000 - 0.0000i
-0.0000 - 0.0000i
  Columns 9 through 12
   0.0000 - 0.0000i  36.0000 + 0.0000i  -0.0000 + 0.0000i
-0.0000 + 0.0000i
  Columns 13 through 16
  -0.0000 + 0.0000i  -0.0000 + 0.0000i  -0.0000 + 0.0000i
-9.0000 + 5.1962i
  Columns 17 through 18
  -0.0000 - 0.0000i  -0.0000 - 0.0000i
```

P5.16

Let $X(k)$ and $Y(k)$ be 10-point DFTs of two 10-point sequences $x(n)$ and $y(n)$, respectively. If

$$X(k) = \exp(j0.2\pi k), \quad 0 \le k \le 9$$

determine $Y(k)$ in each of the following cases without computing the DFT.

1. $y(n) = x((n-5))_{10}$
2. $y(n) = x((n+4))_{10}$
3. $y(n) = x((3-n))_{10}$
4. $y(n) = x(n)e^{j3\pi n/5}$
5. $y(n) = x(n)$ (10) $x((-n))_{10}$

Verify your answers using MATLAB.

Solutions

1. $y(n) = x((n-5))_{10}$: Circular shift by 5. Matlab script:

```
% P5.16
%% P0516a.m
clear;
k = 0:9; X = exp(j*0.2*pi*k); N = length(X);
m = 5; WN = exp(-j*2*pi/N); Y = X.*WN.^(m*k)
% verification
x = real(idft(X,N)); y = cirshftt(x,m,N); Y1 = dft(y,N)
difference = abs(max(Y-Y1))

Y =
```

```
  Columns 1 through 4
   1.0000 + 0.0000i  -0.8090 - 0.5878i   0.3090 + 0.9511i
0.3090 - 0.9511i
  Columns 5 through 8
  -0.8090 + 0.5878i   1.0000 + 0.0000i  -0.8090 - 0.5878i
0.3090 + 0.9511i
  Columns 9 through 10
   0.3090 - 0.9511i  -0.8090 + 0.5878i
Y1 =
  Columns 1 through 4
   1.0000 + 0.0000i  -0.8090 - 0.5878i   0.3090 + 0.9511i
0.3090 - 0.9511i
  Columns 5 through 8
  -0.8090 + 0.5878i   1.0000 + 0.0000i  -0.8090 - 0.5878i
0.3090 + 0.9511i
  Columns 9 through 10
   0.3090 - 0.9511i  -0.8090 + 0.5878i
difference =
   2.6185e-15
```

2. $y(n) = x((n+4))_{10}$: Circular shift by -4. Matlab script:

```
%% P0516b.m
clear;
k = 0:9; X = exp(j*0.2*pi*k); N = length(X);
m = -4; WN = exp(-j*2*pi/N); Y = X.*WN.^(m*k)
% verification
x = real(idft(X,N)); y = cirshftt(x,m,N); Y1 = dft(y,N)
difference = abs(max(Y-Y1))

Y =
  Columns 1 through 4
   1.0000 + 0.0000i  -1.0000 + 0.0000i   1.0000 - 0.0000i
-1.0000 + 0.0000i
  Columns 5 through 8
   1.0000 - 0.0000i  -1.0000 + 0.0000i   1.0000 - 0.0000i
-1.0000 + 0.0000i
  Columns 9 through 10
   1.0000 - 0.0000i  -1.0000 + 0.0000i
Y1 =
  Columns 1 through 4
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i
-1.0000 - 0.0000i
  Columns 5 through 8
   1.0000 + 0.0000i  -1.0000 - 0.0000i   1.0000 + 0.0000i
-1.0000 + 0.0000i
```

```
  Columns 9 through 10
   1.0000 - 0.0000i  -1.0000 + 0.0000i
difference =
   2.2205e-15
```

3. $y(n) = x((3 - n))_{10}$: Circular-fold and circular-shift by 3. Matlab script:

```
%% P0516c.m
clear;
k = 0:9; X = exp(j*0.2*pi*k); N = length(X);
Y  =  circfold(X,N);  m  =  3;  WN  =  exp(-j*2*pi/N);  Y  =
X.*WN.^(m*k)
% verification
x = real(idft(X,N)); y = circfold(x,N); y = cirshftt(x,m,N);
Y1 = dft(y,N)
difference = abs(max(Y-Y1))

Y =
  Columns 1 through 4
   1.0000 + 0.0000i   0.3090 - 0.9511i  -0.8090 - 0.5878i
-0.8090 + 0.5878i
  Columns 5 through 8
   0.3090 + 0.9511i   1.0000 + 0.0000i   0.3090 - 0.9511i
-0.8090 - 0.5878i
  Columns 9 through 10
  -0.8090 + 0.5878i   0.3090 + 0.9511i
Y1 =
  Columns 1 through 4
   1.0000 + 0.0000i   0.3090 - 0.9511i  -0.8090 - 0.5878i
-0.8090 + 0.5878i
  Columns 5 through 8
   0.3090 + 0.9511i   1.0000 + 0.0000i   0.3090 - 0.9511i
-0.8090 - 0.5878i
  Columns 9 through 10
  -0.8090 + 0.5878i   0.3090 + 0.9511i
difference =
   2.6876e-15
```

4. $y(n) = x(n)e^{j3\pi n/5}$: Circular shift in the freq-domain by 3. Matlab script:

```
%% P0516d.m
clear;
k = 0:9; X = exp(j*0.2*pi*k); N = length(X); l = 3;
Y = cirshftt(X,l,N)
% verification
x = real(idft(X,N)); n = 0:9; WN = exp(-j*2*pi/N);
```

```
y = x.*WN.^(-l*n); Y1 = dft(y,N)
difference = abs(max(Y-Y1))

Y =
  Columns 1 through 4
  -0.3090 - 0.9511i   0.3090 - 0.9511i   0.8090 - 0.5878i
1.0000 + 0.0000i
  Columns 5 through 8
   0.8090 + 0.5878i   0.3090 + 0.9511i  -0.3090 + 0.9511i
-0.8090 + 0.5878i
  Columns 9 through 10
  -1.0000 + 0.0000i  -0.8090 - 0.5878i
Y1 =
  Columns 1 through 4
  -0.3090 - 0.9511i   0.3090 - 0.9511i   0.8090 - 0.5878i
1.0000 + 0.0000i
  Columns 5 through 8
   0.8090 + 0.5878i   0.3090 + 0.9511i  -0.3090 + 0.9511i
-0.8090 + 0.5878i
  Columns 9 through 10
  -1.0000 + 0.0000i  -0.8090 - 0.5878i
difference =
   3.2487e-15
```

5. $y(n) = x(n)(10)x((-n))_{10}$: Circular convolution with circularly-folded sequence. Matlab script:

```
%% P0516e.m
clear;
k = 0:9; X = exp(1i*0.2*pi*k); N = length(X);
Y = circfold(X,N); Y = X.*Y
% verification
x = real(idft(X,N)); y = circfold(x,N); y = circonvt(x,y,N);
Y1 = dft(y,N)
difference = abs(max(Y-Y1))

Y =
  Columns 1 through 4
   1.0000 - 0.0000i   1.0000 - 0.0000i   1.0000 + 0.0000i
1.0000 - 0.0000i
  Columns 5 through 8
   1.0000 - 0.0000i   1.0000 - 0.0000i   1.0000 - 0.0000i
1.0000 + 0.0000i
  Columns 9 through 10
   1.0000 - 0.0000i   1.0000 + 0.0000i
Y1 =
```

```
  Columns 1 through 4
   1.0000 + 0.0000i   1.0000 - 0.0000i   1.0000 - 0.0000i
1.0000 + 0.0000i
  Columns 5 through 8
   1.0000 + 0.0000i   1.0000 + 0.0000i   1.0000 - 0.0000i
1.0000 - 0.0000i
  Columns 9 through 10
   1.0000 + 0.0000i   1.0000 + 0.0000i
difference =
   3.4425e-15
```

P5.17

The first six values of the 10-point DFT of a real-valued sequence *x*(*n*) are given by

$$\{10, -2 + j3, 3 + j4, 2 - j3, 4 + j5, 12\}$$

Determine the DFT of each of the following sequences using DFT properties

1. $x_1(n) = x((2 - n))_{10}$
2. $x_2(n) = x((n + 5))_{10}$
3. $x_3(n) = x(n)x((-n))_{10}$
4. $x_4(n) = x(n)$ (10) $x((-n))_{10}$
5. $x_5(n) = x(n)e^{-j4\pi n/5}$

Solutions

1. $x_1(n) = x((2 - n))_{10}$: Circular-folding followed by circ-shifting by 2. Matlab script:

```
% P5.17
%% P0517a.m
%  x(n) is a real-valued sequence  so X(k) = X*((-k))N
clear;clc;
N = 10; X = [10,-2+j*3,3+j*4,2-j*3,4+j*5,12];
X = [X,conj(X(5:-1:2))];x = real(idft(X,N))
WN = exp(-j*2*pi/N); k = 0:N-1; m = 2;
X1 = circfold(X,N); X1 = (WN.^(m*k)).*X1
% Matlab Verification
x1 = circfold(x,N); x1 = cirshftt(x1,m,N); X12 = dft(x1,N)
difference = max(abs(X1-X12))

x =
  Columns 1 through 8
    3.6000   -2.2397    1.0721   -1.3951    3.7520    1.2000
0.6188    1.4132
  Columns 9 through 10
```

```
    1.9571    0.0217
X1 =
  Columns 1 through 4
  10.0000 + 0.0000i  -3.4712 + 0.9751i  -4.7782 + 1.4727i
-3.3814 - 1.2515i
  Columns 5 through 8
   5.9914 + 2.2591i  12.0000 + 0.0000i   5.9914 - 2.2591i
-3.3814 + 1.2515i
  Columns 9 through 10
  -4.7782 - 1.4727i  -3.4712 - 0.9751i
X12 =
  Columns 1 through 4
  10.0000 + 0.0000i  -3.4712 + 0.9751i  -4.7782 + 1.4727i
-3.3814 - 1.2515i
  Columns 5 through 8
   5.9914 + 2.2591i  12.0000 + 0.0000i   5.9914 - 2.2591i
-3.3814 + 1.2515i
  Columns 9 through 10
  -4.7782 - 1.4727i  -3.4712 - 0.9751i
difference =
   1.2429e-14
```

2. $x_2(n) = x((n + 5))_{10}$: 10-point circular shifting by -5. Matlab script:

```
%% P0517b.m
clear;
N = 10; X = [10,-2+j*3,3+j*4,2-j*3,4+j*5,12];
X = [X,conj(X(5:-1:2))]; x = real(idft(X,N))
WN = exp(-j*2*pi/N); k = 0:N-1; m = -5;
X2 = (WN.^(m*k)).*dft(x,N)
% Matlab verification
x2 = cirshftt(x,m,N); X22 = dft(x2,N)
difference = max(abs(X2-X22))
```

```
x =
  Columns 1 through 8
    3.6000   -2.2397    1.0721   -1.3951    3.7520    1.2000
0.6188    1.4132
  Columns 9 through 10
    1.9571    0.0217
X2 =
  Columns 1 through 4
  10.0000 + 0.0000i   2.0000 - 3.0000i   3.0000 + 4.0000i
-2.0000 + 3.0000i
  Columns 5 through 8
```

```
   4.0000 + 5.0000i -12.0000 + 0.0000i   4.0000 - 5.0000i
-2.0000 - 3.0000i
  Columns 9 through 10
   3.0000 - 4.0000i   2.0000 + 3.0000i
X22 =
  Columns 1 through 4
  10.0000 + 0.0000i   2.0000 - 3.0000i   3.0000 + 4.0000i
-2.0000 + 3.0000i
  Columns 5 through 8
   4.0000 + 5.0000i -12.0000 - 0.0000i   4.0000 - 5.0000i
-2.0000 - 3.0000i
  Columns 9 through 10
   3.0000 - 4.0000i   2.0000 + 3.0000i
difference =
   3.2150e-14
```

3. $x_3(n) = x(n)x((-n))_{10}$: Multiplication by circularly-folded sequence. Matlab script:

```
%% P0517c.m
clear;
N = 10; X = [10,-2+j*3,3+j*4,2-j*3,4+j*5,12];
X = [X,conj(X(5:-1:2))]; x = real(idft(X,N))
X3 = circfold(X,N);
X3_ = circonvt(X,X3,N)/10
% Matlab verification
x3 = circfold(x,N); x3 = x.*x3; X32 = dft(x3,N)
difference = max(abs(X3_-X32))

x =
  Columns 1 through 8
    3.6000   -2.2397    1.0721   -1.3951    3.7520    1.2000
0.6188    1.4132
  Columns 9 through 10
    1.9571    0.0217
X3_ =
  Columns 1 through 4
  19.2000 + 0.0000i  10.2000 - 0.0000i  15.6000 + 0.0000i
6.4000 + 0.0000i
  Columns 5 through 8
  10.8000 - 0.0000i  24.4000 + 0.0000i  10.8000 + 0.0000i
6.4000 - 0.0000i
  Columns 9 through 10
  15.6000 - 0.0000i  10.2000 - 0.0000i
X32 =
  Columns 1 through 4
```

```
  19.2000 + 0.0000i  10.2000 - 0.0000i  15.6000 - 0.0000i
6.4000 + 0.0000i
  Columns 5 through 8
  10.8000 - 0.0000i  24.4000 + 0.0000i  10.8000 + 0.0000i
6.4000 - 0.0000i
  Columns 9 through 10
  15.6000 + 0.0000i  10.2000 + 0.0000i
difference =
   3.0681e-14
```

4. $x_4(n) = x(n) ⑩ x((-n))_{10}$: 10-point circular convolution with a circularly-folded sequence. MATLAB script:

```
%% P0517d.m
clear;
N = 10; n = [0:N-1]; X = [10,-2+j*3,3+j*4,2-j*3,4+j*5,12];
X = [X,conj(X(5:-1:2))]; x = real(idft(X,N))
X0 = X(mod(-n,N)+1); X4 = X .* X0
% Matlab Verification
x4 = circonvt(x,x(mod(-n,N)+1),N); X42 = dft(x4,N)
difference = max(abs(X4-X42))

x =
  Columns 1 through 8
    3.6000   -2.2397    1.0721   -1.3951    3.7520    1.2000
0.6188    1.4132
  Columns 9 through 10
    1.9571    0.0217
X4 =
   100    13    25    13    41   144    41    13    25    13
X42 =
   1.0e+02 *
  Columns 1 through 4
   1.0000 + 0.0000i   0.1300 + 0.0000i   0.2500 - 0.0000i
0.1300 - 0.0000i
  Columns 5 through 8
   0.4100 - 0.0000i   1.4400 + 0.0000i   0.4100 + 0.0000i
0.1300 + 0.0000i
  Columns 9 through 10
   0.2500 + 0.0000i   0.1300 - 0.0000i
difference =
   2.5593e-13
```

5. $x_5(n) = x(n)e^{-j4\pi n/5}$: Circular-shifting by -4 in the frequency-domain. Matlab script:

```
%% P0517e.m
clear;
N = 10; n = [0:N-1]; X = [10,-2+j*3,3+j*4,2-j*3,4+j*5,12];
m = 4;
X = [X,conj(X(5:-1:2))]; x = real(idft(X,N))
X5 = [X(m+1:end),X(1:m)]
% Verification
WN = exp(-j*2*pi/N); x5 = x.*(WN.^(m*n)); X51 = dft(x5,N)
difference = max(abs(X5-X51))

x =
  Columns 1 through 8
    3.6000   -2.2397    1.0721   -1.3951    3.7520    1.2000
0.6188    1.4132
  Columns 9 through 10
    1.9571    0.0217
X5 =
  Columns 1 through 4
   4.0000 + 5.0000i  12.0000 + 0.0000i   4.0000 - 5.0000i
2.0000 + 3.0000i
  Columns 5 through 8
   3.0000 - 4.0000i  -2.0000 - 3.0000i  10.0000 + 0.0000i
-2.0000 + 3.0000i
  Columns 9 through 10
   3.0000 + 4.0000i   2.0000 - 3.0000i
X51 =
  Columns 1 through 4
   4.0000 + 5.0000i  12.0000 + 0.0000i   4.0000 - 5.0000i
2.0000 + 3.0000i
  Columns 5 through 8
   3.0000 - 4.0000i  -2.0000 - 3.0000i  10.0000 + 0.0000i
-2.0000 + 3.0000i
  Columns 9 through 10
   3.0000 + 4.0000i   2.0000 - 3.0000i
difference =
   2.4905e-14
```

P5.18

Complex-valued N-point sequence $x(n)$ can be decomposed into N-point circular-conjugatesymmetric and circular-conjugate-antisymmetric sequences using the following relations

$$x_{ccs}(n) \triangleq \frac{1}{2}\left[x(n) + x^*((-n))_N\right]$$

$$x_{cca}(n) \triangleq \frac{1}{2}\left[x(n) - x^*((-n))_N\right]$$

If $X_R(k)$ and $X_I(k)$ are the real and imaginary parts of the N-point DFT of $x(n)$, then

$$\text{DFT}\left[x_{ccs}(n)\right] = X_R(k) \quad \text{and} \quad \text{DFT}\left[x_{cca}(n)\right] = jX_I(k)$$

1. Prove these relations property analytically.
2. Modify the **circevod** function developed in the chapter so that it can be used for complex-valued sequences.
3. Let $X(k) = [3\cos(0.2\pi k) + j4\sin(0.1\pi k)][u(k) - u(k - 20)]$ be a 20-point DFT. Verify this symmetry property using your **circevod** function.

Solutions

1. Using the DFT properties of conjugation and circular folding, we obtain

$$\begin{aligned}
\text{DFT}\left[x_{ccs}(n)\right] &= \frac{1}{2}\left\{\text{DFT}\left[x(n)\right] + \text{DFT}\left[x^*((-n))_N\right]\right\} \\
&= \frac{1}{2}\left\{X(k) + \hat{X}^*((-k))_N\right\}, \text{ where } \hat{X}(k) = \text{DFT}\left[x((-n))_N\right] \\
&= \frac{1}{2}\left\{X(k) + X^*(k)\right\} = \text{Re}\left[X(k)\right] = X_R(k)
\end{aligned}$$

similarly, we can show that

$$\text{DFT}\left[x_{cca}(n)\right] = j\,\text{Im}\left[X(k)\right] = jX_I(k)$$

2. The modified **circevod** function:

```
% P5.18
% Let X(k) = [3 cos(0.2¦Ðk) + j4 sin(0.1¦Ðk)][u(k) ?
u(k ? 20)] be a 20-point DFT. Matlab verification:
N = 20; k = 0:N-1; X = 3*cos(0.2*pi*k) + j*sin(0.1*pi*k);
n = 0:N-1; x = idft(X,N); [xccs, xcca] = circevod(x);
Xccs = dft(xccs,N); Xcca = dft(xcca,N);
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
%
'paperunits','inches','paperposition',[0,0,6,4],'color',[
0,0,0]);
set(Hf_1,'NumberTitle','off','Name','P5.18.3');
subplot(2,2,1); H_s1 = stem(n,real(X),'filled');
set(H_s1,'markersize',3);
title('X_R(k)'); ylabel('Amplitude'); axis([-0.5,20.5,-
4,4]);
subplot(2,2,3); H_s2 = stem(n,real(Xccs),'filled');
```

```
set(H_s2,'markersize',3);
title('X_{ccs}(k)'); ylabel('Amplitude'); xlabel('k');
axis([-0.5,20.5,-4,4]);
subplot(2,2,2); H_s3 = stem(n,imag(X),'filled');
set(H_s3,'markersize',3);
title('X_I(k)'); ylabel('Amplitude'); axis([-0.5,20.5,-
1.1,1.1]);
subplot(2,2,4); H_s4 = stem(n,imag(Xcca),'filled');
set(H_s4,'markersize',3);
title('X_{cca}(k)'); ylabel('Amplitude'); xlabel('k');
axis([-0.5,20.5,-1.1,1.1]);
print -deps2 ../epsfiles/P0518
```

The plots are shown in Figure 5.13.

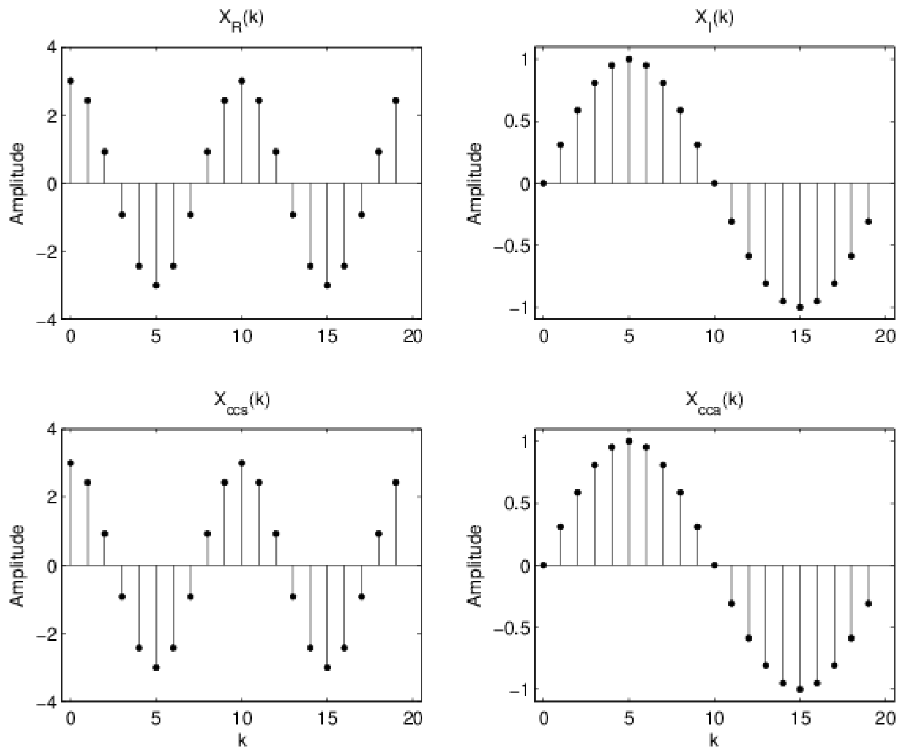


Figure 5.13: Plots in Problem P5.18.3

P5.19

If $X(k)$ is the N-point DFT of an N-point complex-valued sequence

$$x(n) = x_{\mathrm{R}}(n) + jx_{\mathrm{I}}(n)$$

where $x_R(n)$ and $x_I(n)$ are the real and imaginary parts of $x(n)$, then

$$\mathrm{DFT}\left[x_{\mathrm{R}}(n)\right] = X_{\mathrm{ccs}}(k) \quad \text{and} \quad \mathrm{DFT}\left[jx_{\mathrm{I}}(n)\right] = X_{\mathrm{cca}}(k)$$

where $X_{ccs}(k)$ and $X_{cca}(k)$ are the circular-even and circular-odd components of $X(k)$ as defined in Problem P5.18.

1. Prove this property analytically.
2. This property can be used to compute the DFTs of two real-valued N-point sequences using one N-point DFT operation. Specifically, let $x_1(n)$ and $x_2(n)$ be two N-point sequences. Then we can form a complex-valued sequence

$$x(n) = x_1(n) + jx_2(n)$$

and use this property. Develop a MATLAB function to implement this approach with the following format.

function [X1,X2] = real2dft(x1,x2,N)
% DFTs of two real sequences
% [X1,X2] = real2dft(x1,x2,N)
% X1 = n-point DFT of x1
% X2 = n-point DFT of x2
% x1 = sequence of length <= N
% x2 = sequence of length <= N
% N = length of DFT

3. Compute and plot the DFTs of the following two sequences using this function.

$$x_1(n) = \cos(0.1\pi n), \; x_2(n) = \sin(0.2\pi n); \quad 0 \le n \le 39$$

Solutions

1. Analytical proof: Consider

$$\begin{aligned} X_R(k) &\triangleq \text{DFT}\left[x_R(n)\right] = \frac{1}{2}\left\{\text{DFT}\left[x(n)\right] + \text{DFT}\left[x^*(n)\right]\right\} \\ &= \frac{1}{2}\left\{X(k) + X^*((-k))_N\right\} \triangleq X_{\text{ccs}} \end{aligned}$$

Similarly

$$\left.\begin{aligned} jX_I(k) &\triangleq \text{DFT}\left[jx_I(n)\right] = \frac{1}{2}\left\{\text{DFT}\left[x(n)\right] - \text{DFT}\left[x^*(n)\right]\right\} \\ &= \frac{1}{2}\left\{X(k) - X^*((-k))_N\right\} \triangleq X_{\text{cca}} \end{aligned}\right\} \Rightarrow X_I(k) = \frac{X_{\text{cca}}(k)}{j} = -jX_{\text{cca}}(k)$$

2. This property can be used to compute the DFTs of two real-valued N-point sequences using one N point DFT operation. Specifically, let $x_1(n)$ and $x_2(n)$ be two N-point sequences. Then we can form a complex-valued sequence

$$x(n) = x_1(n) + jx_2(n)$$

and use the above property. Matlab function **real2dft**:

```
function [X1,X2] = real2dft(x1,x2,N)
% DFTs of two real sequences
% [X1,X2] = real2dft(x1,x2,N)
% X1 = N-point DFT of x1
```

```
% X2 = N-point DFT of x2
% x1 = real-valued sequence of length <= N
% x2 = real-valued sequence of length <= N
% N = length of DFT
%
% Check for length of x1 and x2
if length(x1) > N
error('*** N must be >= the length of x1 ***')
end
if length(x2) > N
error('*** N must be >= the length of x2 ***')
end
N1 = length(x1); x1 = [x1 zeros(1,N-N1)];
N2 = length(x2); x2 = [x2 zeros(1,N-N2)];
x = x1 + j*x2;
X = dft(x,N);
[X1, X2] = circevod(X); X2 = X2/j;
```

We will also need the **circevod** function for complex sequences (see Problem P5.18). This can be obtained from the one given in the text by two simple changes.

```
function [xccs, xcca] = circevod(x)
% % signal decomposition into circular-even and circular-
odd parts
% % -----------------------------------------------------
---------
% % [xec, xoc] = circevod(x)
% %
% if any(imag(x) ~= 0)
% error('x is not a real sequence')
% end
% N = length(x); n = 0:(N-1);
% xec = 0.5*(x + x(mod(-n,N)+1)); xoc = 0.5*(x - x(mod(-
n,N)+1));

% Complex-valued signal decomposition into circular-even
and circular-odd
% parts-------------------------------------------------
-----------
% [xccs,xcca] = circecvod(x)
%
N = length(x);n = 0:(N-1);
xccs = 0.5*(x + conj(x(mod(-n,N)+1)));
xcca = 0.5*(x - conj(x(mod(-n,N)+1)));
```

3. Compute and plot the DFTs of the following two sequences using the above function

$$x_1(n) = \cos(0.1\pi n),\ x_2(n) = \sin(0.2\pi n); \quad 0 \le n \le 39$$

Matlab verification:

```
% P5.19
% Matlab verification:
N = 40; n = 0:N-1; x1 = cos(0.1*pi*n); x2 =
sin(0.2*pi*n);
[X1,X2] = real2dft(x1,x2,N);
X11 = dft(x1,N); X21 = dft(x2,N);
difference = max(abs(X1-X11))
difference = max(abs(X2-X21))

difference =
   3.6557e-13
difference =
   3.6486e-13
```

P5.20

Using the frequency domain approach, devise a MATLAB function to determine a circular shift $x((n - m))N$, given an $N1$-point sequence $x(n)$ where $N1 \le N$. Your function should have the following format.

```
function y = cirshftf(x,m,N)
% Circular shift of m samples wrt size N in sequence x: (freq domain)
% -----------------------------------------------------------------
% y = cirshftf(x,m,N)
% y : output sequence containing the circular shift
% x : input sequence of length <= N
% m : sample shift
% N : size of circular buffer
% Method: y(n) = idft(dft(x(n))*WN^(mk))
%
% If m is a scalar then y is a sequence (row vector)
% If m is a vector then y is a matrix, each row is a circular shift
% in x corresponding to entries in vecor m
% M and x should not be matrices
```

Verify your function on the following sequence

$$x(n) = \{\underset{\uparrow}{5}, 4, 3, 2, 1, 0, 0, 1, 2, 3, 4\}, \quad 0 \le n \le 10$$

with (a) $m = -5$, $N = 12$ and (b) $m = 8$, $N = 15$.

Circular shifting: The Matlab routine cirshftf.m to implement circular shift is written using the frequency domain property

$$y(n) \triangleq x((n-m))_N = \mathrm{IDFT}\left[X(k)W_N^{mk}\right]$$

This routine will be used in the next problem to generate a circulant matrix and has the following features. If *m* is a scaler then *y*(*n*) is circularly shifted sequence (or array). If *m* is a vector then *y*(*n*) is a matrix, each row of which is a circular shift in *x*(*n*) corresponding to entries in the vector *m*.

```
function y = cirshftf(x,m,N)
% Circular shift of m samples wrt size N in sequence x:
(freq domain)
% -------------------------------------------------------
------------
% function y=cirshftf(x,m,N)
% y : output sequence containing the circular shift
% x : input sequence of length <= N
% m : sample shift
% N : size of circular buffer
%
% Method: y(n) = idft(dft(x(n))*WN^(mk))
%
% If m is a scalar then y is a sequence (row vector)
% If m is a vector then y is a matrix, each row is a
circular shift
% in x corresponding to entries in vecor m
% M and x should not be matrices
%
% Check whether m is scalar, vector, or matrix
[Rm,Cm] = size(m);
if Rm > Cm
m = m'; % make sure that m is a row vector
end
[Rm,Cm] = size(m);
if Rm > 1
error('*** m must be a vector ***') % stop if m is a
matrix
end
% Check whether x is scalar, vector, or matrix
[Rx,Cx] = size(x);
if Rx > Cx
```

```
x = x'; % make sure that x is a row vector
end
[Rx,Cx] = size(x);
if Rx > 1
error('*** x must be a vector ***') % stop if x is a
matrix
end
% Check for length of x
if length(x) > N
error('N must be >= the length of x')
end
x=[x zeros(1,N-length(x))];
X=dft(x,N);
X=ones(Cm,1)*X;
WN=exp(-2*j*pi/N);
k=[0:1:N-1];
Y=(WN.^(m' * k)).*X;

if any(imag(x) ~=0)
y=conj(dfs(conj(Y),N))/N;
% imagy = imag(y);
% [ipos,jpos] = find(imag(y)<eps) ;
% n = [ipos,jpos];
% % y1(n(1),n(2))= real(y(n(1),n(2)));
% for k = 1:length(ipos)
% y(ipos(k),jpos(k))= real(y(ipos(k),jpos(k)));
% end
else
 y=real(conj(dfs(conj(Y),N)))/N;
end

% if any(imag(x) ~=0)
% pos = zeros(1,length(x));
% pos = find(abs(imag(x))<eps);
%
% y(pos)=real(conj(dfs(conj(Y(pos)),N)))/N;

% else
%  y=real(conj(dfs(conj(Y),N)))/N;
% end
```

Matlab verification:

```
% P5.20
% Matlab verification:
```

```
x = [5,4,3,2,1,0,0,1,2,3,4,5];
m = -5; N = 12;
y = cirshftf(x,m,N); y = real(y)

y =
  Columns 1 through 8
   -0.0000   -0.0000    1.0000    2.0000    3.0000    4.0000
5.0000    5.0000
  Columns 9 through 12
    4.0000    3.0000    2.0000    1.0000


% Matlab verification:
x = [5,4,3,2,1,0,0,1,2,3,4,5];
m = 8; N = 15;
y = cirshftf(x,m,N); y = real(y)

y =
  Columns 1 through 8
    1.0000    2.0000    3.0000    4.0000    5.0000    0.0000
0.0000    0.0000
  Columns 9 through 15
    5.0000    4.0000    3.0000    2.0000    1.0000    0.0000
0.0000
```

P5.21

Using the analysis and synthesis equations of the DFT, show that the energy of a sequence satisfies

$$\mathcal{E}_X \triangleq \sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X(k)|^2$$

This is commonly referred to as a *Parseval's relation for the DFT*. Verify this relation using MATLAB on the sequence in Problem P5.20.

Solutions

Parseval's relation for the DFT:

$$\begin{aligned}\sum_{n=0}^{N-1}|x(n)|^2 &= \sum_{n=0}^{N-1}x(n)\,x^*(n)=\sum_{n=0}^{N-1}\left\{\frac{1}{N}\sum_{k=0}^{N-1}X(k)\,W_N^{-nk}\right\}x^*(n)\\ &= \frac{1}{N}\sum_{k=0}^{N-1}X(k)\left\{\sum_{n=0}^{N-1}x^*(n)\,W_N^{-nk}\right\}=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\left\{\sum_{n=0}^{N-1}x(n)\,W_N^{nk}\right\}^*\end{aligned}$$

Therefore,

$$\sum_{n=0}^{N-1}|x(n)|^2=\frac{1}{N}\sum_{k=0}^{N-1}X(k)\,X^*(k)=\frac{1}{N}\sum_{k=0}^{N-1}|X(k)|^2$$

Matlab verification:

```
% P5.21
% Matlab verification:
x = [5,4,3,2,1,0,0,1,2,3,4,5]; N = length(x);
% power of x(n) in the time-domain
power_x = sum(x.*conj(x))
% Power in the frequency-domain
X = dft(x,N); power_X = (1/N)*sum(X.*conj(X))
power_x =
   110
power_X =
  110.0000
```

P5.22

A 512-point DFT $X(k)$ of a real-valued sequence $x(n)$ has the following DFT values:

$X(0)=20+j\alpha$; $X(5)=20+j30$; $X(k_1)=-10+j15$; $X(152)=17+j23$;
$X(k_2)=20-j30$; $X(k_3)=17-j23$; $X(480)=-10-j15$; $X(256)=30+j\beta$

and all other values are known to be zero.

1. Determine the real-valued coefficients α and β.
2. Determine the values of the integers k_1, k_2, and k_3.
3. Determine the energy of the signal $x(n)$.
4. Express the sequence $x(n)$ in a closed form.

Solutions

1. The real-valued coefficients α and β: Since the sequence $x(n)$ is real-valued, $X(k)$ is conjugate symmetric which means that $X(0)$ and $X(N/2)$ are also real-valued. Since $N = 512$, $X(0)$ and $X(256)$ are real-valued. Hence $\alpha = \beta = 0$.
2. The values of the integers k_1, k_2, and k_3: Again using the conjugate symmetry property of the DFT, we have $X(k) = X^*(N - k)$. Thus

$$X(5) = 20 + j30 = X^*(512 - 5) = X^*(507) \Rightarrow X(507) = 20 - j30 \Rightarrow k_2 = 507$$
$$X(480) = -10 - j15 = X^*(512 - 480) = X^*(32) \Rightarrow X(32) = -10 + j15 \Rightarrow k_1 = 32$$
$$X(152) = 17 + j23 = X^*(512 - 152) = X^*(360) \Rightarrow X(360) = 17 - j23 \Rightarrow k_3 = 360$$

3. The energy of the signal $x(n)$: Using Parseval's relation,

$$\begin{aligned}
\mathcal{E}_x &= \sum_{n=-\infty}^{\infty} |x(n)|^2 = \frac{1}{N} \sum_{k=-\infty}^{\infty} |X(k)|^2 \\
&= \frac{1}{512} \left[|X(0)|^2 + 2|X(5)|^2 + 2|X(32)|^2 + 2|X(152)|^2 + |X(256)|^2 \right] = 12.082
\end{aligned}$$

4. Sequence $x(n)$ in a closed form: The time-domain sequence $x(n)$ is a linear combination of the harmonically related complex exponential. Hence

$$\begin{aligned}
x(n) &= \frac{1}{512} \left[X(0) + X(5)e^{-2\pi 5n/512} + X^*(5)e^{2\pi 5n/512} + X(32)e^{-2\pi 32n/512} + X^*(32)e^{2\pi 32n/512} \right. \\
&\quad \left. + X(152)e^{-2\pi 152n/512} + X^*(152)e^{2\pi 152n/512} + X(256)e^{-2\pi 256n/512} \right] \\
&= \frac{1}{512} \left[X(0) + 2\mathrm{Re}\left\{X(5)e^{-2\pi 5n/512}\right\} + 2\mathrm{Re}\left\{X(32)e^{-2\pi 32n/512}\right\} + 2\mathrm{Re}\left\{X(152)e^{-2\pi 152n/512}\right\} \right. \\
&\quad \left. + X(256)(-1)^n \right] \\
&= \frac{1}{512} \left[20 + 72.111 \cos(0.019531\pi n - 56.32^\circ) + 36.056 \cos(0.125\pi n - 123.69^\circ) \right. \\
&\quad \left. + 57.201 \cos(0.59375\pi n - 53.531^\circ) + 30(-1)^n \right]
\end{aligned}$$

P5.23

Let $x(n)$ be a finite length sequence given by

$$x(n) = \left\{ \ldots, 0, 0, 0, \underset{\uparrow}{1}, 2, -3, 4, -5, 0, \ldots \right\}$$

Determine and sketch the sequence $x((-8 - n))_7 R_7(n)$ where

$$\mathcal{R}_7(n) = \begin{cases} 1, & 0 \le n \le 6 \\ 0, & \text{elsewhere} \end{cases}$$

Solutions

Let $x(n)$ be a finite length sequence given by

$$x(n) = \left\{ \ldots, 0, 0, 0, \underset{\uparrow}{1}, 2, -3, 4, -5, 0, \ldots \right\}$$

Then the sequence

$$\begin{aligned}
x((-8 - n))_7 \mathcal{R}_7(n) &= x((-[n + 8]))_7 \mathcal{R}_7(n) = x((-[n + 8 - 7]))_7 \mathcal{R}_7(n) \\
&= x((-[n + 1]))_7 \mathcal{R}_7(n)
\end{aligned}$$

Where

$$\mathcal{R}_7(n) = \begin{cases} 1, & 0 \le n \le 6 \\ 0, & \text{else} \end{cases}$$

is a circularly folded and circularly shifted-by-(-1) version of the 7-point sequence {1, 2, -3, 4, -5, 0, 0}. Hence

$$x((-8-n))_7\mathcal{R}_7(n) = \{0, 0, -5, 4, -3, 2, 1\}$$

P5.24

The **circonvt** function developed in this chapter implements the circular convolution as a matrix-vector multiplication. The matrix corresponding to the circular shifts $\{\mathbf{x}((\mathbf{n}-\mathbf{m}))_N;$ $0 \le n \le N - 1\}$ has an interesting structure. This matrix is called a *circulant* matrix, which is a special case of Toeplitz matrix introduced in Chapter 2.

1. Consider the sequences given in Example 5.13. Express $x_1(n)$ as a column vector $\mathbf{x_1}$ and $x_2((n-m))N$ as a circulant matrix $\mathbf{X_2}$ with rows corresponding to n = 0, 1, 2, 3. Characterize this matrix $\mathbf{X_2}$. Can it completely be described by its first row (or column)?
2. Determine the circular convolution as $\mathbf{X_2x_1}$ and verify your calculations.

Solutions

Circular convolution using circulant matrix operation

$$x_1(n) = \{1, 2, 2\}, \; x_2(n) = \{1, 2, 3, 4\}, \; x_3(n) \triangleq x_1(n) \text{④} x_2(n)$$

1. Using the results from Example 5.13, we can express the above signals as

$$x_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 0 \end{bmatrix}, \quad X2 = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

The matrix $\mathbf{X_2}$ has the property that its every row or column can be obtained from the previous row or column using circular shift. Such a matrix is called a *circulant* matrix. It is completely described by the first column or the row

2. Circular convolution:

$$x_3 = X_2 x_1 = \begin{bmatrix} 1 & 4 & 3 & 2 \\ 2 & 1 & 4 & 3 \\ 3 & 2 & 1 & 4 \\ 4 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \\ 0 \end{bmatrix} = \begin{bmatrix} 15 \\ 12 \\ 9 \\ 14 \end{bmatrix}$$

P5.25

Develop a MATLAB function to construct a circulant matrix **C** given an N-point sequence $x(n)$.

Use the **toeplitz** function to implement matrix **C**. Your subroutine function should have the following format:

function [C] = circulnt(x,N)
% Circulant Matrix from an N-point sequence
% [C] = circulnt(x,N)
% C = circulant matrix of size NxN
% x = sequence of length <= N
% N = size of circulant matrix

Using this function, modify the circular convolution function **circonvt** discussed in the chapter so that the for...end loop is eliminated. Verify your functions on the sequences in Problem P5.24.

Solutions

Matlab function **circulnt**:

```
function C = circulnt(x,N)
% Circulant matrix generation using vector data values
% ----------------------------------------------------
% function C = circulnt(h,N)
%
% C : Circulant matrix
% x : input sequence of length <= N
% N : size of the circular buffer
% Method: C = h((n-m) mod N);
Mx = length(x); % length of x
x = [x, zeros(N-Mx,1)]; % zero-pad x
C = zeros(N,N); % establish size of C
m = 0:N-1; % indices n and m
x = circfold(x,N); % Circular folding
C = cirshft(x,m,N); % Circular shifting
% C = circshift(x,m,N);
```

Matlab verification on sequences in Problem 5.24:

```
% P5.25
% Matlab verification on sequences in Problem 5.24:
clear;
N = 4; x1 = [1,2,2,0]
x2 = [1,2,3,4];
X2 = circulnt(x2,N)
x3 = (X2*x1')'

X2 =
```

```
    1.0000    4.0000    3.0000    2.0000
    2.0000    1.0000    4.0000    3.0000
    3.0000    2.0000    1.0000    4.0000
    4.0000    3.0000    2.0000    1.0000
x3 =
    15    12     9    14
```

P5.26

Using the frequency domain approach, devise a MATLAB function to implement the circular convolution operation between two sequences. The format of the sequence should be

function x3 = circonvf(x1,x2,N)
% Circular convolution in the frequency domain
% x3 = circonvf(x1,x2,N)
% x3 = convolution result of length N
% x1 = sequence of length <= N
% x2 = sequence of length <= N
% N = length of circular buffer

Using your function, compute the circular convolution $\{4, 3, 2, 1\}$ ④ $\{1, 2, 3, 4\}$.

Solutions

Matlab function **circonvf**:

```
function y = circonvf(x1,x2,N)
%
%function y=circonvf(x1,x2,N)
%
% N-point circular convolution between x1 and x2: (freq
domain)
% -------------------------------------------------------
------
% y : output sequence containing the circular convolution
% x1 : input sequence of length N1 <= N
% x2 : input sequence of length N2 <= N
% N : size of circular buffer
%
% Method: y(n) = idft(dft(x1)*dft(x2))
% Check for length of x1
if length(x1) > N
error('N must be >= the length of x1')
end
```

```
% Check for length of x2
if length(x2) > N
error('N must be >= the length of x2')
end
x1=[x1 zeros(1,N-length(x1))];
x2=[x2 zeros(1,N-length(x2))];
X1=fft(x1); X2=fft(x2);
if any(imag(x1) ~=0) || any(imag(x2)~=0)
y=ifft(X1.*X2);
else
y=real(ifft(X1.*X2));
end
```

Circular convolution $\{4, 3, 2, 1\}$ (4) $\{1, 2, 3, 4\}$:

```
% P5.26
x1 = [4,3,2,1];
x2 = [1,2,3,4];
x3 = circonvf(x1,x2,4)
x3_ = circonvt(x1,x2,4);
diff = x3-x3_

x3 =
    24    22    24    30
diff =
     0     0     0     0
```

P5.27

The following four sequences are given:

$$x_1(n) = \{\underset{\uparrow}{1}, 3, 2, -1\}; \; x_2(n) = \{\underset{\uparrow}{2}, 1, 0, -1\}; \; x_3(n) = x_1(n) * x_2(n); \; x_4(n) = x_1(n) \; (5) \; x_2(n)$$

1. Determine and sketch $x_3(n)$.
2. Using $x_3(n)$ alone, determine and sketch $x_4(n)$. Do not directly compute $x_4(n)$.

Solutions

1. Linear convolution $x_3(n)$:

$$x_3(n) = x_1(n) * x_2(n) = \{\underset{\uparrow}{2}, 7, 7, -1, -4, -2, 1\}$$

2. Computation of $x_4(n)$ using $x_3(n)$ alone: The error in the two convolutions is given by

$$e(n) \triangleq x_4(n) - x_3(n) = x_3(n + N)$$

we have, for $N = 5$,

$$e(0) = x_4(0) - x_3(0) = x_3(5) \Rightarrow x_4(0) = x_3(0) + x_3(5) = 2 - 2 = 0$$
$$e(1) = x_4(1) - x_3(1) = x_3(6) \Rightarrow x_4(1) = x_3(1) + x_3(6) = 7 + 1 = 8$$
$$e(2) = x_4(2) - x_3(2) = x_3(7) \Rightarrow x_4(2) = x_3(2) + x_3(7) = 7 + 0 = 7$$
$$e(3) = x_4(3) - x_3(3) = x_3(8) \Rightarrow x_4(3) = x_3(3) + x_3(8) = -1 + 0 = -1$$
$$e(4) = x_4(4) - x_3(4) = x_3(9) \Rightarrow x_4(4) = x_3(4) + x_3(9) = -4 + 0 = -4$$

P5.28

Compute the N-point circular convolution for the following sequences. Plot their samples.

1. $x_1(n) = \sin(\pi n/3)\mathcal{R}_6(n), \quad x_2(n) = \cos(\pi n/4)\mathcal{R}_8(n); \quad N = 10$
2. $x_1(n) = \cos(2\pi n/N)\,\mathcal{R}_N(n), \quad x_2(n) = \sin(2\pi n/N)\,\mathcal{R}_N(n); \quad N = 32$
3. $x_1(n) = (0.8)^n\,\mathcal{R}_N(n), \quad x_2(n) = (-0.8)^n\,\mathcal{R}_N(n); \quad N = 20$
4. $x_1(n) = n\mathcal{R}_N(n), \quad x_2(n) = (N-n)\,\mathcal{R}_N(n); \quad N = 10$
5. $x_1(n) = (0.8)^n R_{20}, \quad x_2(n) = u(n) - u(n-40); \quad N = 50$

Solutions

1. $x_1(n) = \sin(\pi n/3)R_6(n)$, $x_2(n) = \cos(\pi n/4)R_8(n)$; $N = 10$: Matlab script:

```
% P5.28
%% P0528a.m
N = 10; n = 0:N-1;n1 = 0:5; x1 = sin(pi*n1/3);
n2 = 0:7; x2 = cos(pi*n2/4); x3 = circonvt(x1,x2,N);
Hf_1 = figure('Units','inches','position',[1,1,5,2],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.28.1');
H_s1 = stem(n,x3,'filled'); set(H_s1,'markersize',3);
title('Circular Convolution
{\itx}_3({\itn})','fontsize',10);
ylabel('Amplitude'); xlabel('{\itn}'); axis([-
1,N,min(x3)-1,max(x3)+1]);
print -deps2 ../epsfiles/P0528a
```

The sample plot is shown in Figure 5.14.

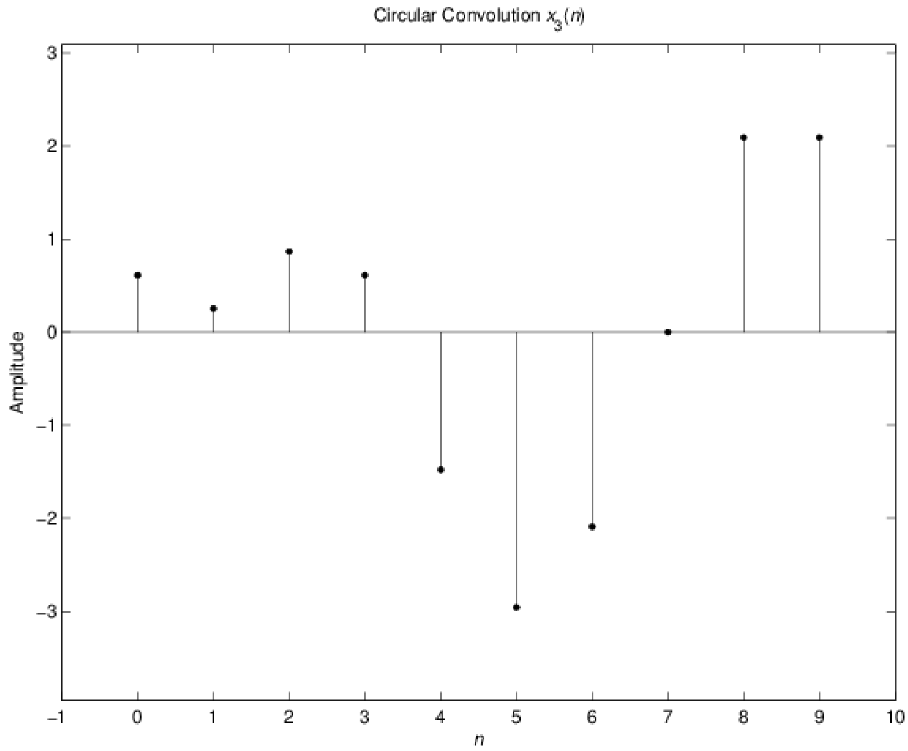


Figure 5.14: The sample plot in Problem P5.28.1

2. $x_1(n) = \cos(2\pi n/N)\ R_N(n)$, $x_2(n) = \sin(2\pi n/N)\ R_N(n)$; $N = 32$: Matlab script

```
%% P0528b.m
N = 32; n = 0:N-1;
x1 = cos(2*pi*n/N); x2 = sin(2*pi*n/N); x3 =
circonvt(x1,x2,N);
Hf_2 = figure('Units','inches','position',[1,1,5,1.5],...
'paperunits','inches');
set(Hf_2,'NumberTitle','off','Name','P5.28.2');
H_s2 = stem(n,x3,'filled'); set(H_s2,'markersize',3);
title('Circular Convolution
{\itx}_3({\itn})','fontsize',10);
ylabel('Amplitude'); xlabel('{\itn}'); axis([-
1,N,min(x3)-1,max(x3)+1]);
print -deps2 ../epsfiles/P0528b
```

The sample plot is shown in Figure 5.15.

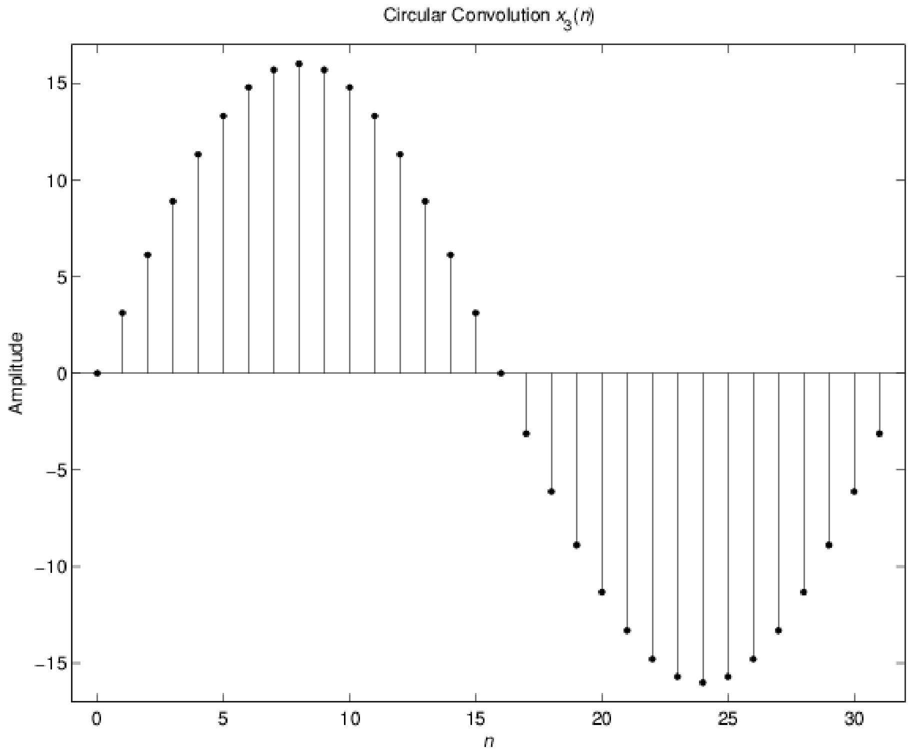


Figure 5.15: The sample plot in Problem P5.28.2

3. $x_1(n) = (0.8)^n R_N(n)$, $x_2(n) = (-0.8)^n R_N(n)$; $N = 20$: Matlab script:

```
%% P0528c.m
N = 20; n = 0:N-1;
x1 = (0.8).^n; x2 = (-0.8).^n; x3 = circonvt(x1,x2,N);
Hf_3 = figure('Units','inches','position',[1,1,5,1.5],...
'paperunits','inches');
set(Hf_3,'NumberTitle','off','Name','P5.28.3');
H_s3 = stem(n,x3,'filled'); set(H_s3,'markersize',3);
title('Circular Convolution
{\itx}_3({\itn})','fontsize',10);
ylabel('Amplitude'); xlabel('{\itn}'); axis([-
1,N,min(x3)-0.5,max(x3)+0.5]);
print -deps2 ../epsfiles/P0528c
```

The sample plot is shown in Figure 5.16.

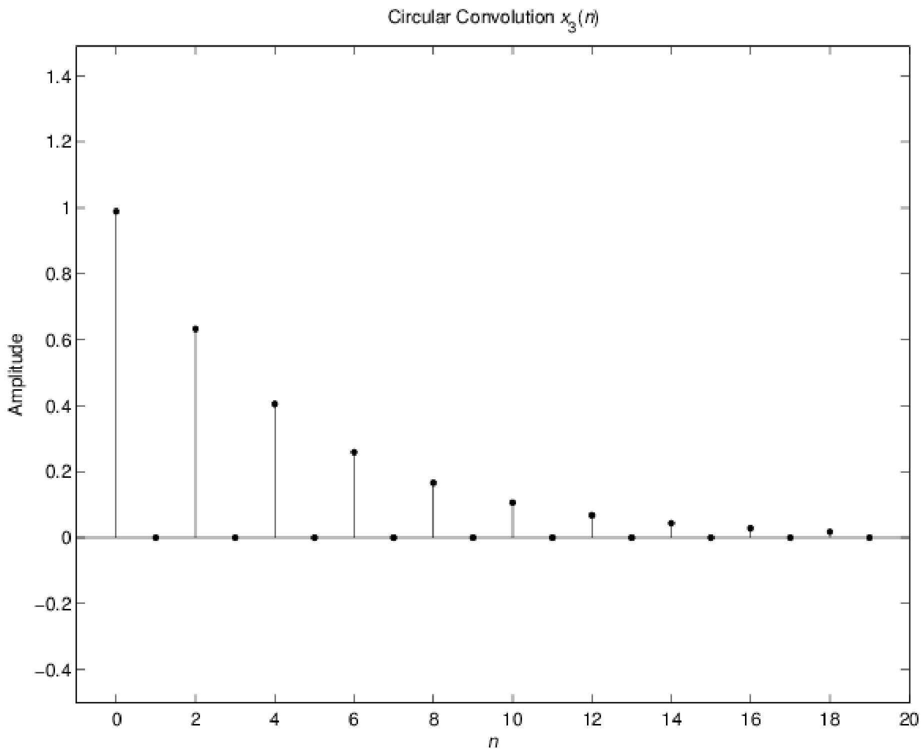


Figure 5.16: The sample plot in Problem P5.28.3

4. $x_1(n) = nR_N(n)$, $x_2(n) = (N - n)R_N(n)$; $N = 10$: Matlab script:

```
%% P0528d.m
N = 10; n = 0:N-1;
x1 = n; x2 = (N-n); x3 = circonvt(x1,x2,N);
Hf_4 = figure('Units','inches','position',[1,1,5,1.5],...
'paperunits','inches');
set(Hf_4,'NumberTitle','off','Name','P5.28.4');
H_s4 = stem(n,x3,'filled'); set(H_s4,'markersize',3);
title('Circular Convolution
{\itx}_3({\itn})','fontsize',10);
ylabel('Amplitude'); xlabel('{\itn}'); axis([-
1,N,min(x3)-10,max(x3)+10]);
print -deps2 ../epsfiles/P0528d
```

The sample plot is shown in Figure 5.17.

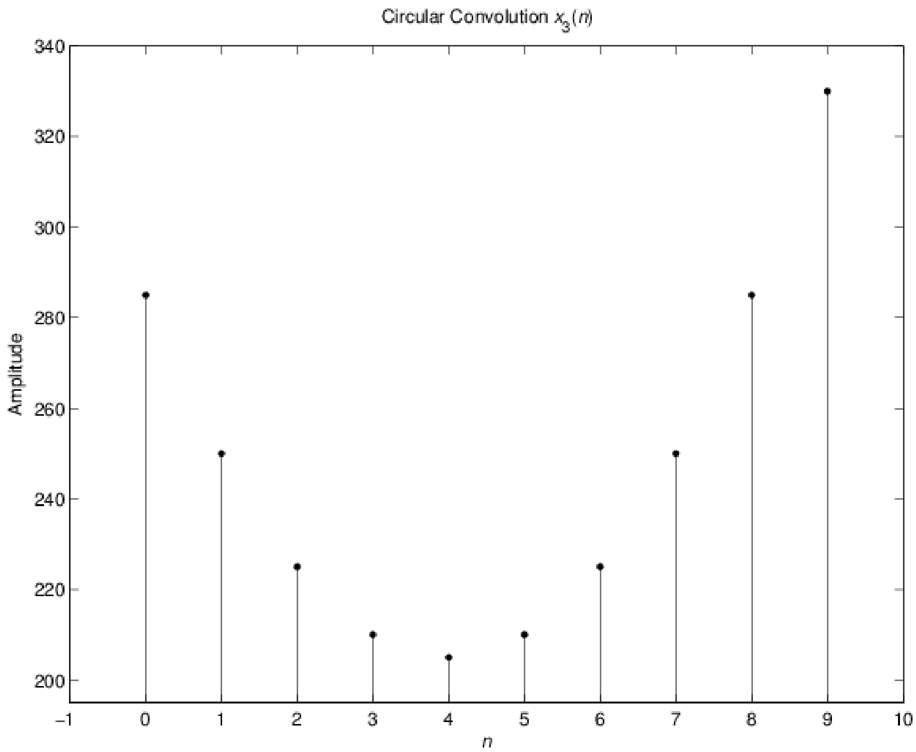


Figure 5.17: The sample plot in Problem P5.28.4

5. $x_1(n) = (0.8)^n R_{20}, x_2(n) = u(n) - u(n - 40)$; $N = 50$: Matlab script:

```
%% P0528e.m
N = 50; n = 0:N-1; n1 = 0:19; x1 = (0.8).^n1;
n2 = 0:39; x2 = ones(1,40); x3 = circonvt(x1,x2,N);
Hf_5 = figure('Units','inches','position',[1,1,5,1.5],...
'paperunits','inches');
set(Hf_5,'NumberTitle','off','Name','P5.28.5');
H_s5 = stem(n,x3,'filled'); set(H_s5,'markersize',3);
title('Circular Convolution
{\itx}_3({\itn})','fontsize',10);
ylabel('Amplitude'); xlabel('{\itn}'); axis([-
1,N,min(x3)-1,max(x3)+1]);
print -deps2 ../epsfiles/P0528e
```

The sample plot is shown in Figure 5.18.

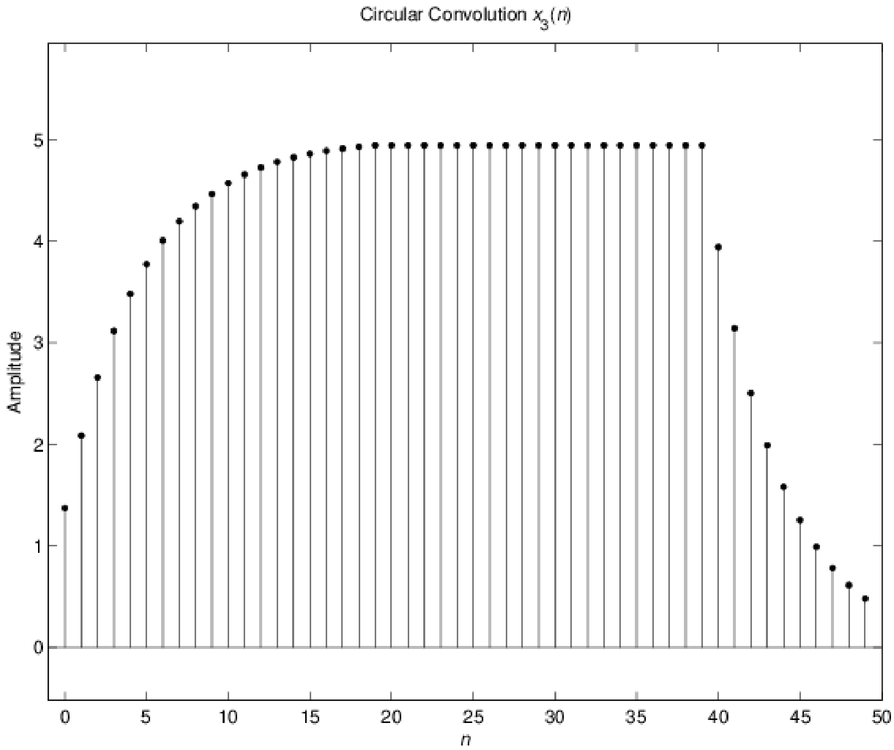


Figure 5.18: The sample plot in Problem P5.28.5

P5.29

Let $x_1(n)$ and $x_2(n)$ be two N-point sequences.

1. If $y(n) = x_1(n) \; \textcircled{N} \; x_2(n)$ show that

$$\sum_{n=0}^{N-1} y(n) = \left(\sum_{n=0}^{N-1} x_1(n) \right) \left(\sum_{n=0}^{N-1} x_2(n) \right)$$

2. Verify this result for the following sequences.

$$x_1(n) = \{9, 4, -1, 4, -4, -1, 8, 3\}; \quad x_2(n) = \{-5, 6, 2, -7, -5, 2, 2, -2\}$$

Solutions

1. Since $y(n) = x_1(n) \textcircled{N} x_2(n) = \sum_{k=0}^{N-1} x_1(k)\, x_2((n-k))_N$ we have

$$\begin{aligned}
\sum_{n=0}^{N-1} y(n) &= \sum_{n=0}^{N-1}\sum_{k=0}^{N-1} x_1(k)\, x_2((n-k))_N = \sum_{k=0}^{N-1} x_1(k) \sum_{n=0}^{N-1} x_2((n-k))_N \\
&= \sum_{n=0}^{N-1} x_1(n) \left[\sum_{n=0}^{k-1} x_2(n-k+N) + \sum_{n=k}^{N-1} x_2(n-k)\right] \\
&= \sum_{n=0}^{N-1} x_1(n) \left[\sum_{n=N-k}^{N-1} x_2(n) + \sum_{n=0}^{N-1-k} x_2(n)\right] = \sum_{n=0}^{N-1} x_1(n) \left[\sum_{n=0}^{N-k-1} x_2(n) + \sum_{n=N-k}^{N-1} x_2(n)\right] \\
&= \left(\sum_{n=0}^{N-1} x_1(n)\right)\left(\sum_{n=0}^{N-1} x_2(n)\right)
\end{aligned}$$

2. Verification using the following sequences:

$$x_1(n) = \{9, 4, -1, 4, -4, -1, 8, 3\};\ \ x_2(n) = \{-5, 6, 2, -7, -5, 2, 2, -2\}$$

Consider

$$x_1(n) = \{9, 4, -1, 4, -4, -1, 8, 3\} \Rightarrow \sum_{n=0}^{7} x_1(n) = 22$$

$$x_2(n) = \{-5, 6, 2, -7, -5, 2, 2, -2\} \Rightarrow \sum_{n=0}^{7} x_2(n) = -7$$

$$y(n) = x_1(n) \textcircled{8} x_2(n) = \{14, -9, -32, -74, -7, -16, -57, 27\} \Rightarrow \sum_{n=0}^{7} y(n) = -154$$

Hence

$$\sum_{n=0}^{7} y(n) = -154 = (22) \times (-7) = \left(\sum_{n=0}^{7} x_1(n)\right)\left(\sum_{n=0}^{7} x_2(n)\right)$$

```
% P5.29
% Matlab Verification
N = 8; n = 0:N-1;
x1 = [-9,4,-1,4,-4,-1,8,3];
x2 = [-5,6,2,-7,-5,2,2,-2];
y1 = sum(circonvt(x1,x2,N));
y2 =sum(x1)*sum(x2);
dif = max(abs(y1-y2))

dif =

   1.2079e-13
```

P5.30

Let $X(k)$ be the 8-point DFT of a 3-point sequence $x(n) = \{5, \underset{\uparrow}{-4}, 3\}$. Let $Y(k)$ be the 8-point DFT of a sequence $y(n)$. Determine $y(n)$ when $Y(k) = \sum_8^{5k} X(-k)_8$

Solutions

Using the circular folding and the circular shifting properties of the DFT, we have

$$y(n) = \text{IDFT}\left[W_8^{5k} X((-k))_8\right] = \text{IDFT}\left[X((-k))_8\right]_{n \to (n-5)}$$
$$= \left[x((-n))_8|_{n \to (n-5)}\right] \mathcal{R}_8(n) = x((5-n))_8 \mathcal{R}_8(n) = \{\underset{\uparrow}{0}, 0, 0, 3, -4, 5, 0, 0\}$$

P5.31

For the following sequences compute (i) the N-point circular convolution $x_3(n) = x_1(n)$ $Ⓝ$ $x_2(n)$, (ii) the linear convolution $x_4(n) = x_1(n) * x_2(n)$, and (iii) the error sequence $e(n) = x_3(n) - x_4(n)$.

1. $x_1(n) = \{1, 1, 1, 1\}$, $\quad x_2(n) = \cos(\pi n/4)\, \mathcal{R}_6(n)$; $\quad N = 8$
2. $x_1(n) = \cos(2\pi n/N)\, \mathcal{R}_{16}(n)$, $\quad x_2(n) = \sin(2\pi n/N)\, \mathcal{R}_{16}(n)$; $\quad N = 32$
3. $x_1(n) = (0.8)^n\, \mathcal{R}_{10}(n)$, $\quad x_2(n) = (-0.8)^n\, \mathcal{R}_{10}(n)$; $\quad N = 15$
4. $x_1(n) = n\mathcal{R}_{10}(n)$, $\quad x_2(n) = (N - n)\, \mathcal{R}_{10}(n)$; $\quad N = 10$
5. $x_1(n) = \{1, -1, 1, -1\}$, $\quad x_2(n) = \{1, 0, -1, 0\}$; $\quad N = 5$

In each case verify that $e(n) = x_4(n + N)$.

Solutions

1. $x_1(n) = \{1, 1, 1, 1\}$, $x_2(n) = \cos(\pi n/4)\mathbf{R}_6(n)$; $N = 8$:

```
% P5.31
%% P0531a.m
clear;clc;close all;
x1 = [1,1,1,1]; x2 = cos(pi*[0:5]/4); N = 8; n3 = 0:N-1;
x3 = circonvt(x1,x2,N);
x4 = conv(x1,x2); n4 = 0:length(x4)-1;
e1 = x3 - x4(1:N); e2 = x4(N+1:end);Ne2 = length(e2);
e2 = [e2,zeros(1,length(n3)-Ne2)];
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.31.1');
subplot(4,1,1); stem(n3,x3,'linewidth',1); axis([-1,9,-
3,2]);
title('Circular Convolution: {\itx}_3({\itn})');
```

```
ylabel('Amplitude');
subplot(4,1,2); stem(n4,x4,'linewidth',1); axis([-1,9,-
3,2]);
title('Linear Convolution: {\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,3); stem(n3,e1,'linewidth',1); axis([-1,9,-
3,2]);
title('Error: {\itx}_3({\itn})-{\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,4); stem(n3,e2,'linewidth',1); axis([-1,9,-
3,2]);
title('Error: {\itx}_4({\itn+N})'); ylabel('Amplitude');
print -deps2 ../epsfiles/P0531a
```

The plots of various signals are shown in Figure 5.19.

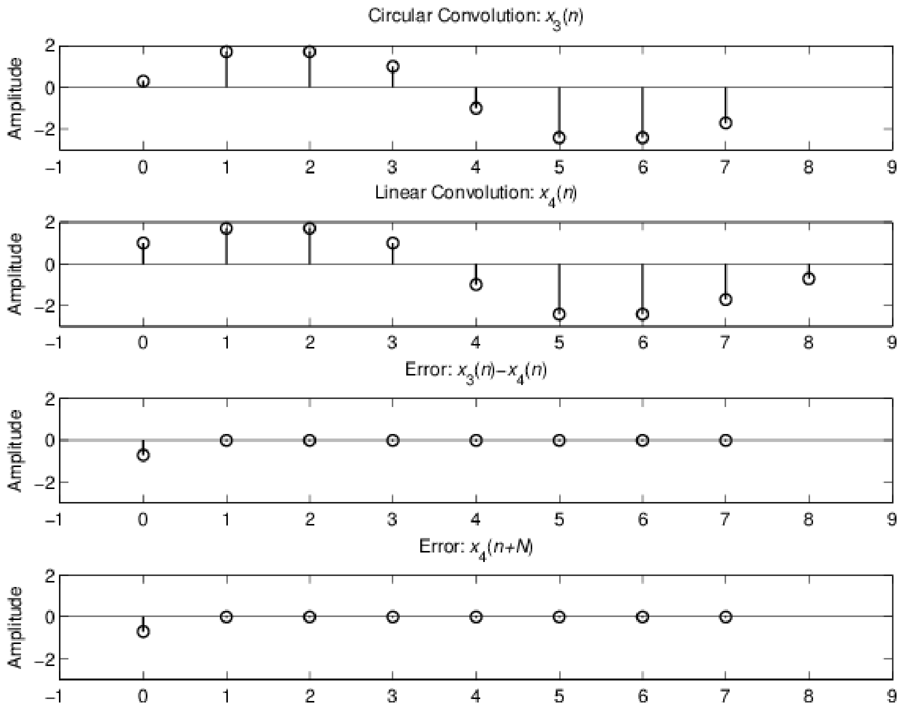


Figure 5.19: The sample plot of various signals in Problem P5.31.1

2. $x_1(n) = \cos(2\pi n/N)\mathbf{R}_{16}(n)$, $x_2(n) = \sin(2\pi n/N)\mathbf{R}_{16}(n)$; $N = 32$:

```
%% P0531b.m
clear;clc;close all;
N = 32; x1 = cos(2*pi*[0:15]/N); x2 = sin(2*pi*[0:15]/N);
x3 = circonvt(x1,x2,N); n3 = 0:N-1;
x4 = conv(x1,x2); n4 = 0:length(x4)-1;
e1 = x3 - [x4,0];
e2 = x4(N+1:end);Ne2 = length(e2);
e2 = [e2,zeros(1,length(n3)-Ne2)];
```

```
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.31.2');
subplot(4,1,1); stem(n3,x3,'linewidth',1); axis([-1,31,-
6,6]);
title('Circular Convolution: {\itx}_3({\itn})');
ylabel('Amplitude');
subplot(4,1,2); stem(n4,x4,'linewidth',1); axis([-1,31,-
6,6]);
title('Linear Convolution: {\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,3); stem(n3,e1,'linewidth',1); axis([-1,31,-
6,6]);
title('Error: {\itx}_3({\itn})-{\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,4); stem(n3,e2,'linewidth',1); axis([-1,31,-
6,6]);
title('Error: {\itx}_4({\itn+N})'); ylabel('Amplitude');
print -deps2 ../epsfiles/P0531b
```

The plots of various signals are shown in Figure 5.20.

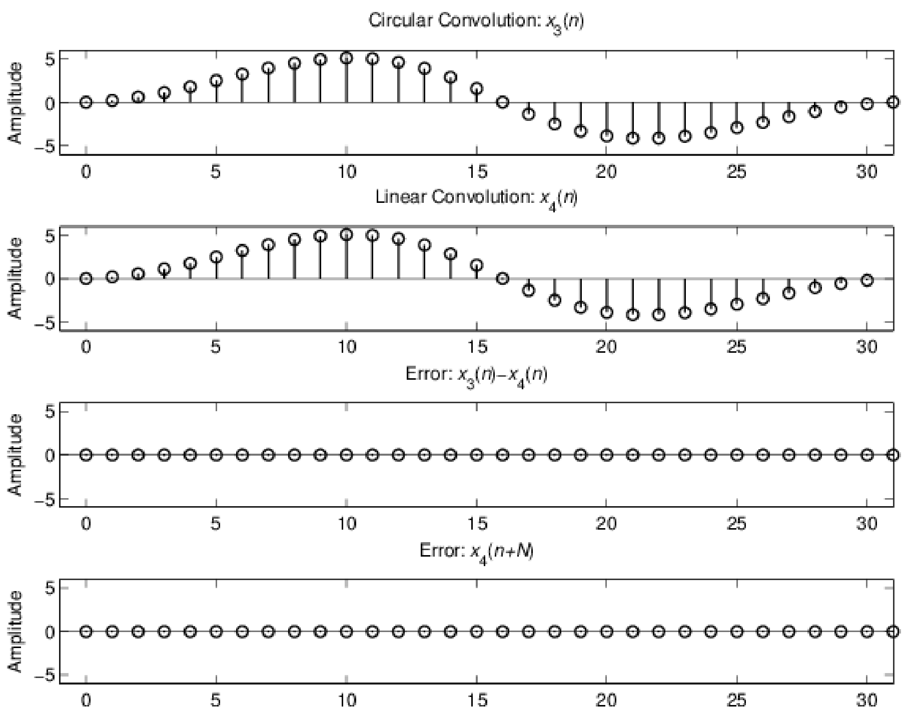


Figure 5.20: The sample plot of various signals in Problem P5.31.2

3. $x_1(n) = (0.8)^n \mathbf{R_{10}}(n)$, $x_2(n) = (-0.8)^n \mathbf{R_{10}}(n)$; $N = 15$:

```
%% P0531c.m
```

```
clear;clc;close all;
N = 15; x1 = (0.8).^[0:9]; x2 = (-0.8).^[0:9];
x3 = circonvt(x1,x2,N); n3 = 0:N-1;
x4 = conv(x1,x2); n4 = 0:length(x4)-1;
e1 = x3 - x4(1:N);
e2 = x4(N+1:end); Ne2 = length(e2);
e2 = [e2,zeros(1,length(n3)-Ne2)];
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.31.2');
subplot(4,1,1); stem(n3,x3,'linewidth',1); axis([-1,19,-
0.5,1.5]);
title('Circular Convolution: {\itx}_3({\itn})');
ylabel('Amplitude');
subplot(4,1,2); stem(n4,x4,'linewidth',1); axis([-1,19,-
0.5,1.5]);
title('Linear Convolution: {\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,3); stem(n3,e1,'linewidth',1); axis([-1,19,-
0.5,1.5]);
title('Error: {\itx}_3({\itn})-{\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,4); stem(n3,e2,'linewidth',1); axis([-1,19,-
0.5,1.5]);
title('Error: {\itx}_4({\itn+N})'); ylabel('Amplitude');
print -deps2 ../epsfiles/P0531c
```

The plots of various signals are shown in Figure 5.21.

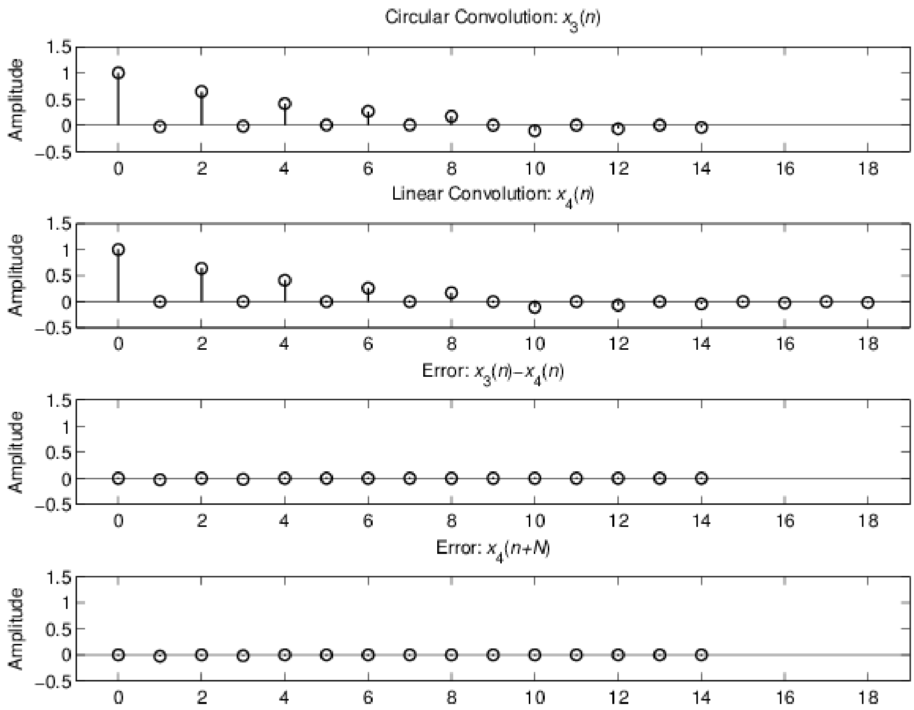


Figure 5.21: The sample plot of various signals in Problem P5.31.3

4. $x_1(n) = n\mathbf{R}_{10}(n)$, $x_2(n) = (N - n)\mathbf{R}_{10}(n)$; $N = 10$:

```
%% P0531d.m
clear;clc;close all;
N = 10; n = 0:N-1; x1 = n; x2 = N-n;
x3 = circonvt(x1,x2,N); n3 = 0:N-1;
x4 = conv(x1,x2); n4 = 0:length(x4)-1;
e1 = x3 - x4(1:N);
e2 = x4(N+1:end); Ne2 = length(e2);
e2 = [e2,zeros(1,length(n3)-Ne2)];
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.31.2');
subplot(4,1,1); stem(n3,x3,'linewidth',1); axis([-
1,19,0,350]);
title('Circular Convolution: {\itx}_3({\itn})');
ylabel('Amplitude');
subplot(4,1,2); stem(n4,x4,'linewidth',1); axis([-
1,19,0,350]);
title('Linear Convolution: {\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,3); stem(n3,e1,'linewidth',1); axis([-
1,19,0,350]);
title('Error: {\itx}_3({\itn})-{\itx}_4({\itn})');
ylabel('Amplitude');
```

```
subplot(4,1,4); stem(n3,e2,'linewidth',1); axis([-
1,19,0,350]);
title('Error: {\itx}_4({\itn+N})'); ylabel('Amplitude');
print -deps2 ../epsfiles/P0531d
```

The plots of various signals are shown in Figure 5.22.

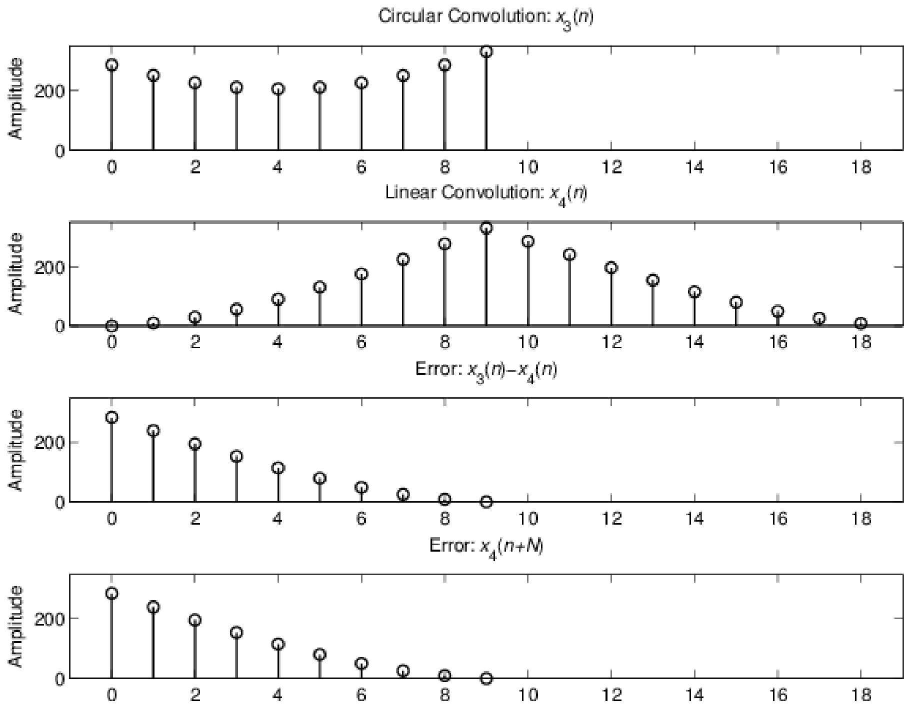


Figure 5.22: The sample plot of various signals in Problem P5.31.4

5. $x_1(n) = \{1, -1, 1, -1\}$, $x_2(n) = \{1, 0, -1, 0\}$; $N = 5$:

```
%% P0531e.m
clear;clc;close all;
N = 5; n = 0:N-1; x1 = [1,-1,1,-1]; x2 = [1,0,-1,0];
x3 = circonvt(x1,x2,N); n3 = 0:N-1;
x4 = conv(x1,x2); n4 = 0:length(x4)-1;
e1 = x3 - x4(1:N);
e2 = x4(N+1:end); Ne2 = length(e2);
e2 = [e2,zeros(1,length(n3)-Ne2)];
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.31.2');
subplot(4,1,1); stem(n3,x3,'linewidth',1); axis([-1,7,-
2.5,2.5]);
title('Circular Convolution: {\itx}_3({\itn})');
ylabel('Amplitude');
subplot(4,1,2); stem(n4,x4,'linewidth',1); axis([-1,7,-
2.5,2.5]);
```

```
title('Linear Convolution: {\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,3); stem(n3,e1,'linewidth',1); axis([-1,7,-
2.5,2.5]);
title('Error: {\itx}_3({\itn})-{\itx}_4({\itn})');
ylabel('Amplitude');
subplot(4,1,4); stem(n3,e2,'linewidth',1); axis([-1,7,-
2.5,2.5]);
title('Error: {\itx}_4({\itn+N})'); ylabel('Amplitude');
print -deps2 ../epsfiles/P0531e
```

The plots of various signals are shown in Figure 5.23.

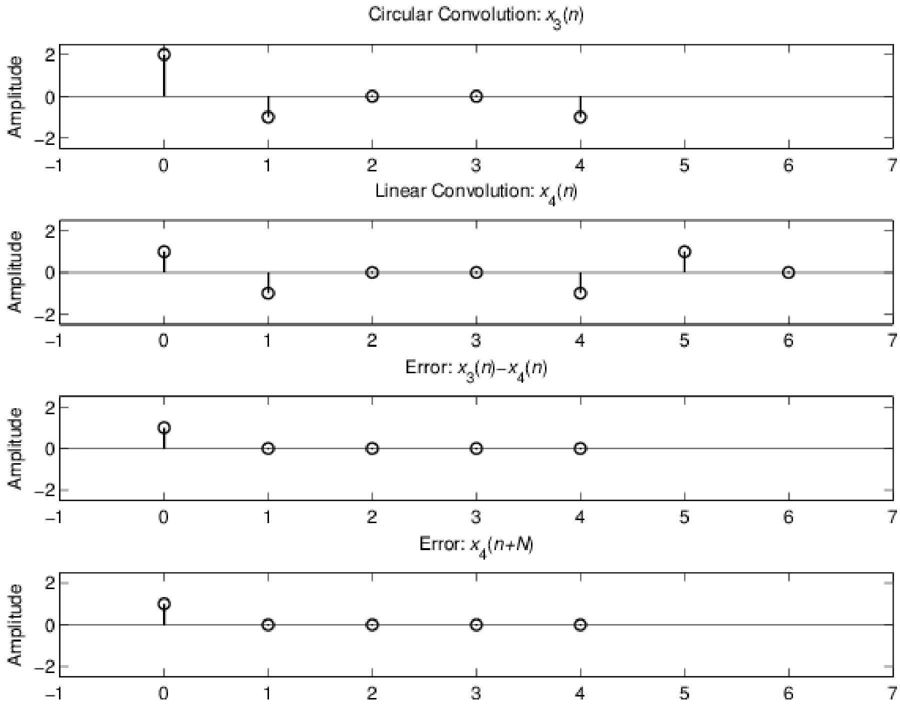


Figure 5.23: The sample plot of various signals in Problem P5.31.5

P5.32

The overlap-add method of block convolution is an alternative to the overlap-save method. Let $x(n)$ be a long sequence of length ML where $M, L \gg 1$. Divide $x(n)$ into M segments $\{x_m(n), m = 1, \ldots, M\}$ each of length L

$$x_m(n) = \begin{cases} x(n), & mL \le n \le (m+1)L - 1 \\ 0, & \text{elsewhere} \end{cases} \quad \text{so that} \quad x(n) = \sum_{m=0}^{M-1} x_m(n)$$

Let $h(n)$ be an L-point impulse response. Then

$$y(n) = x(n) * h(n) = \sum_{m=0}^{M-1} x_m(n) * h(n) = \sum_{m=0}^{M-1} y_m(n); \quad y_m(n) \triangleq x_m(n) * h(n)$$

Clearly, $y_m(n)$ is a $(2L - 1)$-point sequence. In this method we have to save the intermediate convolution results and then properly overlap these before adding to form the final result $y(n)$. To use DFT for this operation we have to choose $N \geq (2L - 1)$.

1. Develop a MATLAB function to implement the overlap-add method using the circular convolution operation. The format should be

```
function [y] = ovrlpadd(x,h,N)
% Overlap-Add method of block convolution
% [y] = ovrlpadd(x,h,N)
%
% y = output sequence
% x = input sequence
% h = impulse response
% N = block length >= 2*length(h)-1
```

2. Incorporate the radix-2 FFT implementation in this function to obtain a high-speed overlap-add block convolution routine. Remember to choose $N = 2v$.

3. Verify your functions on the following two sequences

$$x(n) = \cos(\pi n/500)\, \mathcal{R}_{4000}(n), \quad h(n) = \{1, -1, 1, -1\}$$

Solutions

1. Matlab function to implement the overlap-add method using the circular convolution operation:

```
function [y] = ovrlpadd(x,h,N)
% Overlap-Add method of block convolution
% ------------------------------------------
% [y] = ovrlpadd(x,h,N)
% y = output sequence
% x = input sequence
% h = impulse response
% N = DFT length >= 2*length(h)-1
%
Lx = length(x); L = length(h); L1 = L-1;
h = [h zeros(1,N-L)];
%
M = ceil(Lx/L); % Number of blocks
x = [x, zeros(1,M*L-Lx)]; % append (M*N-Lx) zeros
Y = zeros(M,N); % Initialize Y matrix
%
% convolution with succesive blocks
for m = 0:M-1
```

```
xm = [x(m*L+1:(m+1)*L),zeros(1,N-L)];
Y(m+1,:) = circonvt(xm,h,N);
end
%
% Overlap and Add
Y = [Y,zeros(M,1)]; Y = [Y;zeros(1,N+1)];
Y1 = Y(:,1:L); Y1 = Y1'; y1 = Y1(:);
Y2 = [zeros(1,L);Y(1:M,L+1:2*L)]; Y2 = Y2'; y2 = Y2(:);
y = y1+y2; y = y'; y = removetrailzeros(y);
```

2. The radix-2 FFT implementation for high-speed block convolution:

```
function [y] = hsolpadd(x,h)
% High-Speed Overlap-Add method of block convolution244
Solutions Manual for DSP using Matlab (2nd Edition) 2006
% ----------------------------------------------------
% [y] = hsolpadd(x,h)
% y = output sequence (real-valued)
% x = input sequence (real-valued)
% h = impulse response (real-valued)
%
Lx = length(x); L = length(h); N = 2^ceil(log2(2*L-1));
H = fft(h,N);
%
M = ceil(Lx/L); % Number of blocks
x = [x, zeros(1,M*L-Lx)]; % append (M*N-Lx) zeros
Y = zeros(M,N); % Initialize Y matrix
%
% convolution with succesive blocks
for m = 0:M-1
xm = [x(m*L+1:(m+1)*L),zeros(1,N-L)];
Y(m+1,:) = real(ifft(fft(xm,N).*H,N));
end
%
% Overlap and Add
Y = [Y,zeros(M,1)]; Y = [Y;zeros(1,N+1)];
Y1 = Y(:,1:L); Y1 = Y1'; y1 = Y1(:);
Y2 = [zeros(1,L);Y(1:M,L+1:2*L)]; Y2 = Y2'; y2 = Y2(:);
y = y1+y2; y = y'; y = removetrailzeros(y);
```

3. Verification using the following two sequences

$$x(n) = \cos(\pi n/500)\,\mathcal{R}_{4000}(n), \quad h(n) = \{1, -1, 1, -1\}$$

```
% P5.32
% Matlab Verification
```

```
n = 0:4000-1; x = cos(pi*n/500); h = [1,-1,1,-1];
y1 = ovrlpadd(x,h,7);
y2 = hsolpadd(x,h);
y3 = conv(x,h);
e1 = max(abs(y1-y3))
e2 = max(abs(y1-y2(1:end-1)))

e1 =
   1.5543e-15
e2 =
   1.7764e-15
```

P5.33

Given the following sequences $x_1(n)$ and $x_2(n)$:

$$x_1(n) = \{2, 1, 1, 2\}, \quad x_2(n) = \{1, -1, -1, 1\}$$

1. Compute the circular convolution $x_1(n) \; N \; x_2(n)$ for $N = 4$, 7, and 8.
2. Compute the linear convolution $x_1(n) * x_2(n)$.
3. Using results of calculations, determine the minimum value of N necessary so that linear and circular convolutions are same on the N-point interval.
4. Without performing the actual convolutions, explain how you could have obtained the result of P5.33.3.

Solutions

1. Circular convolutions $x_1(n) \; N \; x_2(n)$:

$$N = 4 \; : x_1(n) \; 4 \; x_2(n) = \{0, 0, 0, 0\}$$
$$N = 7 \; : x_1(n) \; 7 \; x_2(n) = \{2, -1, 2, 0, -2, 1, -2\}$$
$$N = 8 \; : x_1(n) \; 8 \; x_2(n) = \{2, -1, 2, 0, -2, 1, -2, 0\}$$

2. The linear convolution: $x_1(n) * x_2(n)$ = {2, -1, 2, 0, -2, 1, -2}.

3. From the results of the above two parts, the minimum value of N to make the circular convolution equal to the linear convolution is 7.

4. If we make N equal to the length of the linear convolution which is equal to the length of $x_1(n)$ plus the length of $x_2(n)$ minus one, then the desired result can be achieved. In this case then $N = 4 + 4 - 1 = 7$, as expected

```
% P5.33
x1 = [2,1,1,2];x2 = [1,-1,-1,1];
%% P5.33.1a
N = 4; n = 0:N-1;
y1 = circonvt(x1,x2,N)
%% P5.33.1b
N = 7; n = 0:N-1;
```

```
y2 = circonvt(x1,x2,N)
%% P5.33.1c
N = 8; n = 0:N-1;
y3 = circonvt(x1,x2,N)
%% P5.33.2
y = conv(x1,x2)

y1 =
   0.0000   -2.0000        0    2.0000
y2 =
   2.0000   -1.0000   -2.0000    2.0000   -2.0000   -1.0000
2.0000
y3 =
   2.0000   -1.0000   -2.0000    2.0000   -2.0000   -1.0000
2.0000   -0.0000
y =
     2    -1    -2     2    -2    -1     2
```

P5.34

Let

$$x(n) = \begin{cases} A\cos(2\pi\ell n/N), & 0 \le n \le N-1 \\ 0, & \text{elsewhere} \end{cases} = A\cos(2\pi\ell n/N)\,\mathcal{R}_N(n)$$

Where ℓ is an integer. Notice that *x*(*n*) contains *exactly* ℓ periods (or cycles) of the cosine waveform in *N* samples. This is a windowed cosine sequence containing *no leakage*.

1. Show that the DFT *X*(*k*) is a real sequence given by

$$X(k) = \frac{AN}{2}\delta(k-\ell) + \frac{AN}{2}\delta(k-N+\ell); \quad 0 \le k \le (N-1), \quad 0 < \ell < N$$

2. Show that if ℓ = 0, then the DFT *X*(*k*) is given by

$$X(k) = AN\delta(k); \quad 0 \le k \le (N-1)$$

3. Explain clearly how these results should be modified if ℓ < 0 or ℓ > *N*.
4. Verify the results of parts 1, 2, and 3 using the following sequences. Plot the real parts of the DFT sequences using the **stem** function.

(a) $x_1(n) = 3\cos(0.04\pi n)\,\mathcal{R}_{200}(n)$
(b) $x_2(n) = 5\mathcal{R}_{50}(n)$
(c) $x_3(n) = [1 + 2\cos(0.5\pi n) + \cos(\pi n)]\,\mathcal{R}_{100}(n)$
(d) $x_4(n) = \cos(25\pi n/16)\,\mathcal{R}_{64}(n)$
(e) $x_5(n) = [4\cos(0.1\pi n) - 3\cos(1.9\pi n)]\,\mathcal{R}_{40}(n)$

Solutions

1. Consider the DFT $X(k)$ of $x(n)$ which is given by

$$
\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} A\cos\left(\frac{2\pi \ell n}{N}\right) e^{-j\frac{2\pi}{N}kn}, \quad 0 \le k \le N-1 \\
&= \frac{A}{2}\sum_{n=0}^{N-1}\left\{e^{j\frac{2\pi}{N}\ell n} + e^{-j\frac{2\pi}{N}\ell n}\right\} e^{-j\frac{2\pi}{N}kn}, \quad 0 \le k \le N-1 \\
&= \frac{A}{2}\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(k-\ell)n} + \frac{A}{2}\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(k+\ell)n}, \quad 0 \le k \le N-1 \\
&= \frac{AN}{2}\delta(k-\ell) + \frac{AN}{2}\delta(k-N+\ell); \quad 0 \le k \le (N-1), \quad 0 < \ell < N
\end{aligned}
$$

which is a real-valued sequence.

2. If $\ell = 0$, then the DFT $X(k)$ is given by

$$
\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} A\, e^{-j\frac{2\pi}{N}kn}, \quad 0 \le k \le N-1 \\
&= AN\delta(k); \quad 0 \le k \le (N-1)
\end{aligned}
$$

3. If $\ell < 0$ or $\ell > N$, then we must replace it by $((\ell))_N$ in the result of part 1., i.e.

$$
X(k) = \frac{AN}{2}\delta\left[k - ((\ell))_N\right] + \frac{AN}{2}\delta\left[k - N + ((\ell))_N\right]; \quad 0 \le k \le (N-1)
$$

4. Verification of the results of parts 1., 2., and 3. above using Matlab and the following sequences:

(a) $x_1(n) = 3\cos(0.04\pi n)\mathbf{R}_{200}(n)$:

```
% P5.34
%% P0534a.m
clear;clc;close all;
N = 200; n = 0:N-1; x1 = 3*cos(0.04*pi*n); l = 4;
k = 0:N-1; X1 = real(fft(x1,N));
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.34.4(a)');
subplot(2,1,1); H_s1 = stem(n,x1,'g','filled');
set(H_s1,'markersize',1);
title('Sequence: {\itx}_1({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-4,4]); xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X1,'r','filled');
set(H_s2,'markersize',2);
title('DFT: {\itX}_1({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-10,310]);
xlabel('\itk');
set(gca,'xtick',[0,l,N-l],'ytick',[0,300])
```

```
print -deps2 ../epsfiles/P0534a
```

The sequence $x_1(n)$ and its DFT $X_1(k)$ are shown in Figure 5.24.

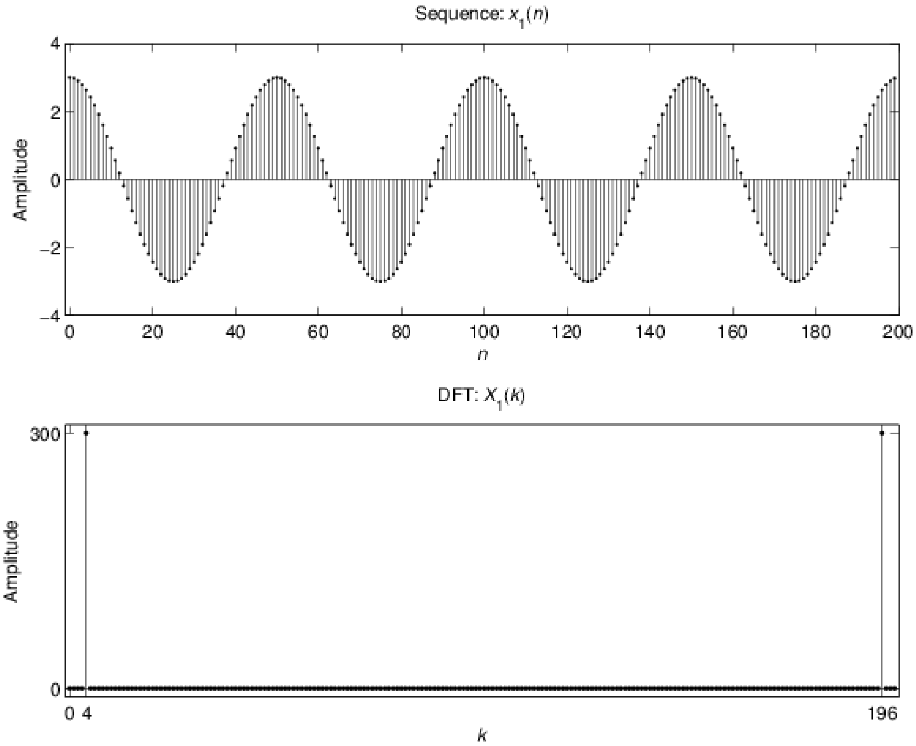


Figure 5.24: The signal $x_1(n)$ and its DFT $X_1(k)$ in Problem P5.34.4(a)

(b) $x_2(n) = 5\mathbf{R}_{50}(n)$:

```
%% P0534b.m
clear;clc;close all;
N = 50; n = 0:N-1; x2 = 5*cos(0*pi*n); l = 0;
k = 0:N-1; X2 = real(fft(x2,N));
Hf_2 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_2,'NumberTitle','off','Name','P5.34.4(b)');
subplot(2,1,1); H_s1 = stem(n,x2,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}_2({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-1,6]); xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X2,'r','filled');
set(H_s2,'markersize',2);
title('DFT: {\itX}_2({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-10,260]);
xlabel('\itk');
set(gca,'xtick',[0,N-1],'ytick',[0,250])
print -deps2 ../epsfiles/P0534b
```

The sequence $x_2(n)$ and its DFT $X_2(k)$ are shown in Figure 5.25.

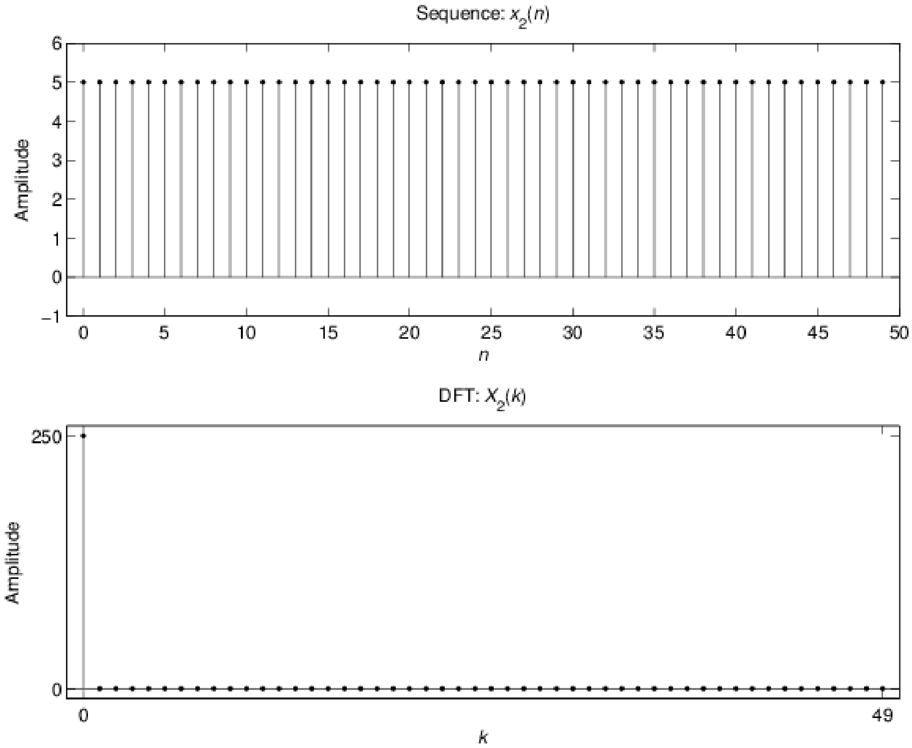


Figure 5.25: The signal $x_1(n)$ and its DFT $X_2(k)$ in Problem P5.34.4(b)

(c) $x_3(n) = [1 + 2\cos(0.5\pi n) + \cos(\pi n)]\mathbf{R}_{100}(n)$:

```
%% P0534c.m
clear;clc;close all;
N = 100; n = 0:N-1; x3 = 1+2*cos(0.5*pi*n)+cos(pi*n); l1
= 0; l2 = 25; l3 = 50;
k = 0:N-1; X3 = real(fft(x3,N));
Hf_3 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_3,'NumberTitle','off','Name','P5.34.4(c)');
subplot(2,1,1); H_s1 = stem(n,x3,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}_3({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-1,5]); xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X3,'r','filled');
set(H_s2,'markersize',2);
title('DFT: {\itX}_3({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-10,110]);
xlabel('\itk');
set(gca,'xtick',[l1,l2,l3,N-l2,N-1],'ytick',[0,100])
print -deps2 ../epsfiles/P0534c
```

The sequence $x_3(n)$ and its DFT $X_3(k)$ are shown in Figure 5.26.

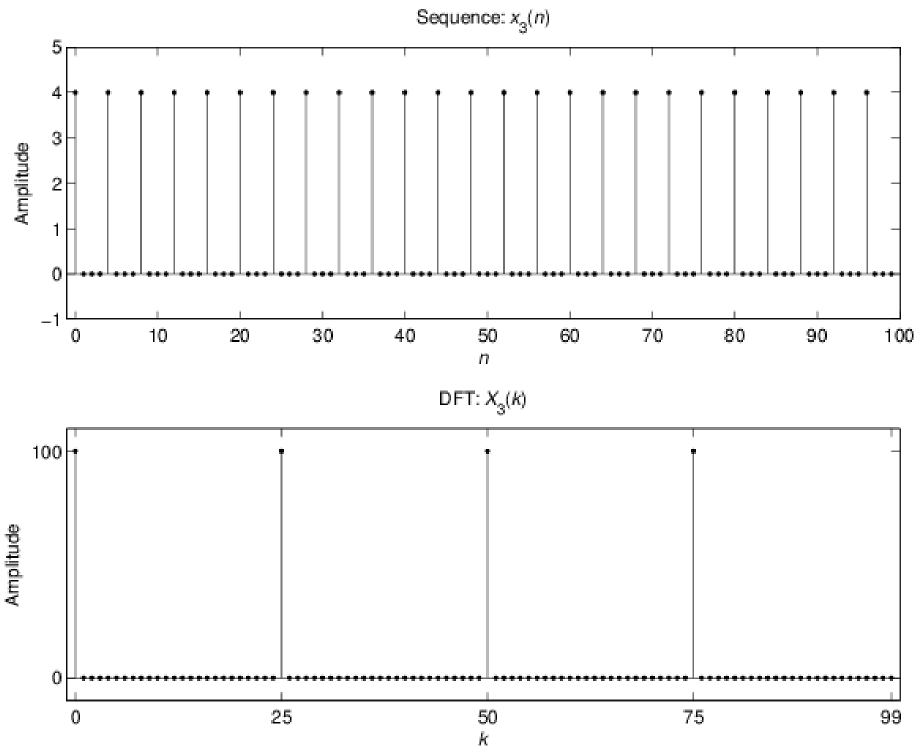


Figure 5.26: The signal $x_3(n)$ and its DFT $X_3(k)$ in Problem P5.34.4(c)

(d) $x_4(n) = \cos(25\pi n/16)\mathbf{R}_{64}(n)$:

```
%% P0534d.m
clear;clc;close all;
N = 64; n = 0:N-1; x4 = cos(25*pi*n/16); l = 50;
k = 0:N-1; X4 = real(fft(x4,N));
Hf_4 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_4,'NumberTitle','off','Name','P5.34.4(d)');
subplot(2,1,1); H_s1 = stem(n,x4,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}_4({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-1.1,1.1]);
xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X4,'r','filled');
set(H_s2,'markersize',2);
title('DFT: {\itX}_4({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-5,35]); xlabel('\itk');
set(gca,'xtick',[0,N-1,l,N-1],'ytick',[0,32])
print -deps2 ../epsfiles/P0534d
```

The sequence $x_4(n)$ and its DFT $X_4(k)$ are shown in Figure 5.27.

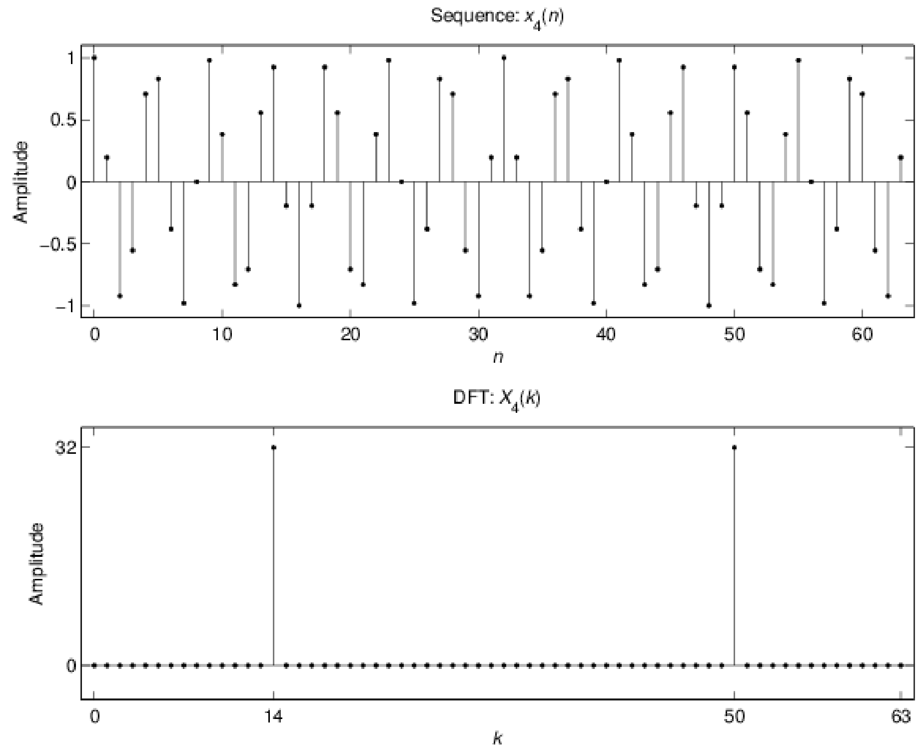


Figure 5.27: The signal $x_4(n)$ and its DFT $X_4(k)$ in Problem P5.34.4(d)

(e) $x_5(n) = [\cos(0.1\pi n) - 3\cos(1.9\pi n)]\mathbf{R_{40}}(n)$:

```
%% P0534e.m
clear;clc;close all;
N = 40; n = 0:N-1; x5 = 4*cos(0.1*pi*n)-3*cos(1.9*pi*n);
l1 = 2; l2 = 38;
k = 0:N-1; X5 = real(fft(x5,N));
Hf_5 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_5,'NumberTitle','off','Name','P5.34.4(e)');
subplot(2,1,1); H_s1 = stem(n,x5,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}_5({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-1.1,1.1]);
xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X5,'r','filled');
set(H_s2,'markersize',2);
title('DFT: {\itX}_5({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-5,25]); xlabel('\itk');
set(gca,'xtick',[0,l1,l2,N],'ytick',[0,20])
print -deps2 ../epsfiles/P0534e
```

The sequence $x_5(n)$ and its DFT $X_5(k)$ are shown in Figure 5.28.

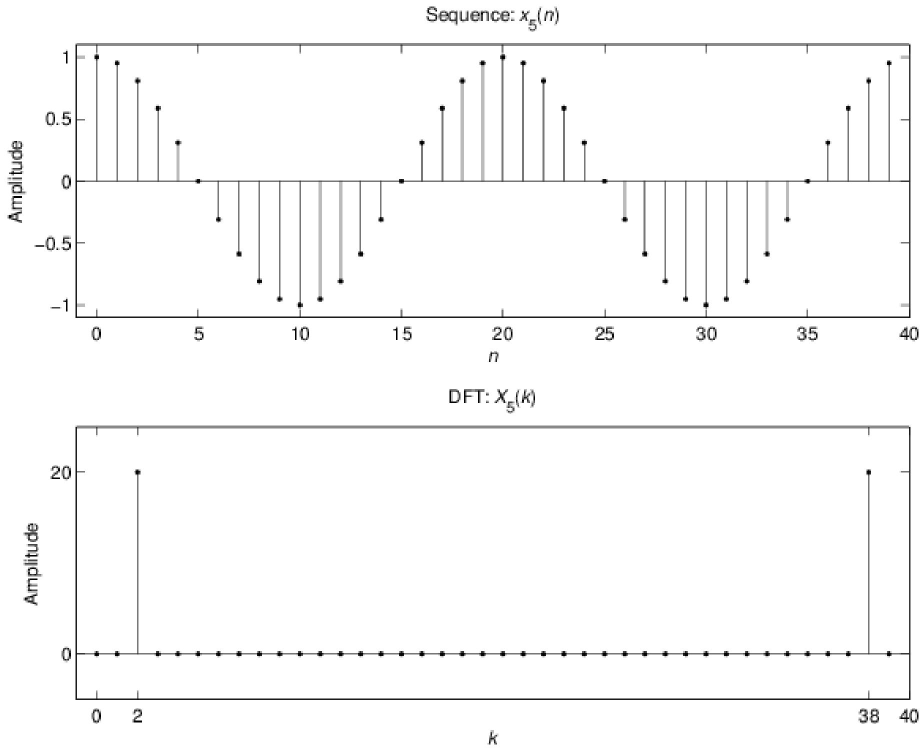


Figure 5.28: The sample plot of various signals in Problem P5.34.4(e)

P5.35

Let $x(n) = A\cos(\omega_0 n)\, R_N(n)$, where ω_0 is a real number.

1. Using the properties of the DFT, show that the real and the imaginary parts of $X(k)$ are given by

$$
\begin{aligned}
X(k) &= X_R(k) + jX_I(k) \\
X_R(k) &= (A/2)\cos\left[\tfrac{\pi(N-1)}{N}(k - f_0N)\right]\frac{\sin[\pi(k - f_0N)]}{\sin[\pi(k - f_0N)/N]} \\
&\quad + (A/2)\cos\left[\tfrac{\pi(N-1)}{N}(k + f_0N)\right]\frac{\sin[\pi(k - N + f_0N)]}{\sin[\pi(k - N + f_0N)/N]} \\
X_I(k) &= -(A/2)\sin\left[\tfrac{\pi(N-1)}{N}(k - f_0N)\right]\frac{\sin[\pi(k - f_0N)]}{\sin[\pi(k - f_0N)/N]} \\
&\quad - (A/2)\sin\left[\tfrac{\pi(N-1)}{N}(k + f_0N)\right]\frac{\sin[\pi(k - N + f_0N)]}{\sin[\pi(k - N + f_0N)/N]}
\end{aligned}
$$

2. This result implies that the original frequency ω_0 of the cosine waveform has *leaked* into other frequencies that form the harmonics of the time-limited sequence, and hence it is called the leakage property of cosines. It is a natural result due to the fact that bandlimited periodic cosines are sampled over noninteger periods. Explain this result using the periodic extension $\tilde{x}(n)$ of $x(n)$ and the result in Problem P5.34.1.

3. Verify the leakage property using $x(n) = \cos(5\pi n/99)\ R_{200}(n)$. Plot the real and the imaginary parts of $X(k)$ using the **stem** function.

Solutions

1. Consider

$$X(k)=\sum_{n=0}^{N-1}x(n)e^{-j\left(\frac{2\pi}{N}\right)nk}=A\sum_{n=0}^{N-1}\cos(\omega_0 n)\left\{\cos\left(\frac{2\pi}{N}nk\right)-j\sin\left(\frac{2\pi}{N}nk\right)\right\}$$

$$X_R(k)+jX_I(k)=A\sum_{n=0}^{N-1}\cos(\omega_0 n)\cos\left(\frac{2\pi}{N}nk\right)-jA\sum_{n=0}^{N-1}\cos(\omega_0 n)\sin\left(\frac{2\pi}{N}nk\right)$$

Hence

$$X_R(k)=A\sum_{n=0}^{N-1}\cos(\omega_0 n)\cos\left(\frac{2\pi}{N}nk\right) \tag{5.5}$$

$$X_I(k)=-A\sum_{n=0}^{N-1}\cos(\omega_0 n)\sin\left(\frac{2\pi}{N}nk\right) \tag{5.6}$$

Consider the real-part in (5.5),

$$\begin{aligned}X_R(k)&=A\sum_{n=0}^{N-1}\cos(\omega_0 n)\cos\left(\frac{2\pi}{N}nk\right)=\frac{A}{2}\sum_{n=0}^{N-1}\left\{\cos\left(\omega_0 n-\frac{2\pi}{N}nk\right)+\cos\left(\omega_0 n+\frac{2\pi}{N}nk\right)\right\}\\&=\frac{A}{2}\sum_{n=0}^{N-1}\left\{\cos\left(2\pi f_0 n-\frac{2\pi}{N}nk\right)+\cos\left(2\pi f_0 n+\frac{2\pi}{N}nk\right)\right\}\qquad [\because \omega_0=2\pi f_0]\\&=\frac{A}{2}\sum_{n=0}^{N-1}\left\{\cos\left[\frac{2\pi}{N}(f_0N-k)n\right]+\cos\left[\frac{2\pi}{N}(f_0N+k)n\right]\right\}\\&=\frac{A}{2}\sum_{n=0}^{N-1}\left\{\cos\left[\frac{2\pi}{N}(k-f_0N)n\right]+\cos\left[\frac{2\pi}{N}\{k-(N-f_0N)\}n\right]\right\},\ 0\le k<N\end{aligned} \tag{5.7}$$

To reduce the sum-of-cosine terms in (5.7), consider

$$\begin{aligned}\sum_{n=0}^{N-1}\cos\left(\frac{2\pi}{N}vn\right)&=\frac{1}{2}\sum_{n=0}^{N-1}e^{j\left(\frac{2\pi}{N}\right)vn}+\frac{1}{2}\sum_{n=0}^{N-1}e^{-j\left(\frac{2\pi}{N}\right)vn}=\frac{1}{2}\left(\frac{1-e^{j2\pi v}}{1-e^{j\frac{2\pi}{N}v}}\right)+\frac{1}{2}\left(\frac{1-e^{-j2\pi v}}{1-e^{-j\frac{2\pi}{N}v}}\right)\\&=\frac{1}{2}e^{-j\pi v\left(\frac{N-1}{N}\right)}\frac{\sin(\pi v)}{\sin(\pi v/N)}+\frac{1}{2}e^{j\pi n\left(\frac{N-1}{N}\right)}\frac{\sin(\pi v)}{\sin(\pi v/N)}\\&=\cos\left\{\frac{\pi v(N-1)}{N}\right\}\frac{\sin(\pi n)}{\sin(\pi v/N)}\end{aligned} \tag{5.8}$$

Now substituting (5.8) in the first term of (5.7) with v = (k - f_0N) and in the second term of (5.7) with v = (k - [N - f_0N]), we obtain the desired result

$$\begin{aligned}X_R(k)&=\frac{A}{2}\cos\left\{\frac{\pi(N-1)}{N}(k-f_0N)\right\}\frac{\sin[\pi(f_0N-k)]}{\sin[\frac{\pi}{N}(f_0N-k)]}\\&+\frac{A}{2}\cos\left\{\frac{\pi(N-1)}{N}(k-[N-f_0N])\right\}\frac{\sin\{\pi(k-[N-f_0N])\}}{\sin[\frac{\pi}{N}(f_0N-k)]}\end{aligned} \tag{5.9}$$

Similarly, we can show that

$$X_1(k) = -\frac{A}{2}\sin\left\{\tfrac{\pi(N-1)}{N}(k - f_0 N)\right\}\frac{\sin[\pi(f_0 N - k)]}{\sin[\frac{\pi}{N}(f_0 N - k)]}$$
$$-\frac{A}{2}\sin\left\{\tfrac{\pi(N-1)}{N}(k - [N - f_0 N])\right\}\frac{\sin\{\pi(k - [N - f_0 N])\}}{\sin[\frac{\pi}{N}(f_0 N - k)]} \qquad (5.10)$$

2. The above result implies that the original frequency ω_0 of the cosine waveform has *leaked* into other frequencies that form the harmonics of the time-limited sequence and hence it is called the leakage property of cosines. It is a natural result due to the fact that bandlimited periodic cosines are sampled over noninteger periods. Due to this fact, the periodic extension of $x(n)$ does not result in a continuation of the cosine waveform but has a jump at every N interval. This jump results in the leakage of one frequency into the abducent frequencies and hence the result of the Problem P5.34.1 do not apply.

3. Verification of the leakage property using $x(n) = \cos(5\pi n/99)\ \mathbf{R}_{200}(n)$:

```
% P5.35
% Matlab Verification
clear;clc;close all;
N = 200; n = 0:N-1; x = cos(5*pi*n/99); I = 5;
k = 0:N-1; X = fft(x,N);
Hf = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
%
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,6,4]);
set(Hf,'NumberTitle','off','Name','P5.35.3');
subplot(3,1,1); H_s1 = stem(n,x,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-1,1]); xlabel('\itn');
% set(gca,'xtick',[0:20:N],'ytick',[-1:0.5:1])
subplot(3,1,2); H_s2 = stem(n,real(X),'r','filled');
set(H_s2,'markersize',2);
title('Real-part of the DFT:
{\itX}_R({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-10,100]);
xlabel('\itk');
set(gca,'xtick',[0,I,N/2,N-I,N],'ytick',[-10,0,100])
subplot(3,1,3); H_s3 = stem(n,imag(X),'g','filled');
set(H_s3,'markersize',2);
title('Imaginary-part of the DFT:
{\itX}_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-20,20]); xlabel('\itk');
set(gca,'xtick',[0,I,N/2,N-I,N],'ytick',[-20,0,20])
print -deps2 ../epsfiles/P0535
```

The sequence $x(n)$, the real-part of its DFT $X_R(k)$, and the imaginary part of its DFT $X_I(k)$ are shown in Figure 5.29.

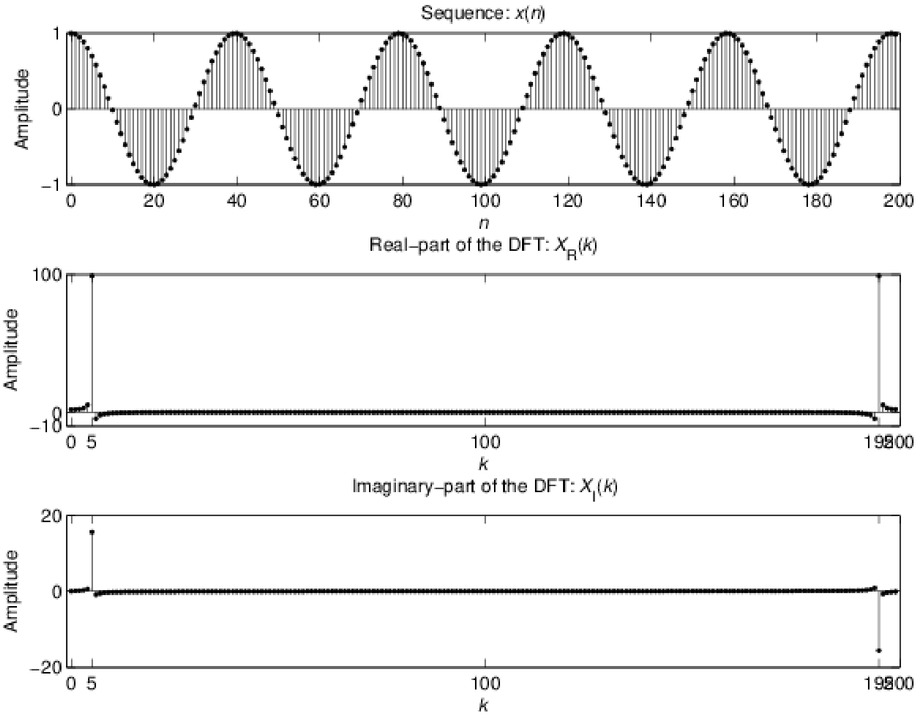


Figure 5.29: The leakage property of a cosine signal in Problem P5.35.3

P5.36

Let

$$x(n) = \begin{cases} A\sin\left(2\pi\ell n/N\right), & 0 \le n \le N-1 \\ 0, & \text{Elsewhere} \end{cases} = A\sin\left(2\pi\ell n/N\right)\mathcal{R}_N(n)$$

where ℓ is an integer. Notice that $x(n)$ contains *exactly* ℓ periods (or cycles) of the sine waveform in N samples. This is a windowed sine sequence containing *no leakage*.

1. Show that the DFT $X(k)$ is a purely imaginary sequence given by

$$X(k) = \frac{AN}{2j}\delta\left(k-\ell\right) - \frac{AN}{2j}\delta(k-N+\ell); \quad 0 \le k \le (N-1), \quad 0 < \ell < N$$

2. Show that if $\ell = 0$, then the DFT $X(k)$ is given by

$$X(k) = 0; \quad 0 \le k \le (N-1)$$

3. Explain clearly how these results should be modified if $\ell < 0$ or $\ell > N$.
4. Verify the results of parts 1, 2, and 3 using the following sequences. Plot the imaginary parts of the DFT sequences using the **stem** function.

(a) $x_1(n) = 3\sin(0.04\pi n)\,\mathcal{R}_{200}(n)$
(b) $x_2(n) = 5\sin 10\pi n \mathcal{R}_{50}(n)$
(c) $x_3(n) = [2\sin(0.5\pi n) + \sin(\pi n)]\,\mathcal{R}_{100}(n)$
(d) $x_4(n) = \sin(25\pi n/16)\,\mathcal{R}_{64}(n)$
(e) $x_5(n) = [4\sin(0.1\pi n) - 3\sin(1.9\pi n)]\,\mathcal{R}_{20}(n)$

Solutions

1. Consider the DFT $X(k)$ of $x(n)$ which is given by

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} A\sin\left(\frac{2\pi\ell n}{N}\right) e^{-j\frac{2\pi}{N}kn}, \quad 0 \le k \le N-1 \\
&= \frac{A}{j2}\sum_{n=0}^{N-1}\left\{e^{j\frac{2\pi}{N}\ell n} - e^{-j\frac{2\pi}{N}\ell n}\right\} e^{-j\frac{2\pi}{N}kn}, \quad 0 \le k \le N-1 \\
&= \frac{A}{j2}\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(k-\ell)n} - \frac{A}{2}\sum_{n=0}^{N-1} e^{-j\frac{2\pi}{N}(k+\ell)n}, \quad 0 \le k \le N-1 \\
&= \frac{AN}{j2}\delta(k-\ell) - \frac{AN}{j2}\delta(k-N+\ell); \quad 0 \le k \le (N-1), \quad 0 < \ell < N
\end{aligned}$$

which is a purely imaginary-valued sequence.

2. If $\ell = 0$, then the DFT $X(k)$ is given by

$$\begin{aligned}
X(k) &= \sum_{n=0}^{N-1} x(n)\, e^{-j\frac{2\pi}{N}kn} = \sum_{n=0}^{N-1} 0\, e^{-j\frac{2\pi}{N}kn}, \quad 0 \le k \le (N-1) \\
&= 0; \quad 0 \le k \le (N-1)
\end{aligned}$$

3. If $\ell < 0$ or $\ell > N$, then we must replace it by $((\ell))_N$ in the result of part 1., i.e.

$$X(k) = \frac{AN}{j2}\delta[k-((\ell))_N] - \frac{AN}{j2}\delta[k-N+((\ell))_N]; \quad 0 \le k \le (N-1)$$

4. Verification of the results of parts 1., 2., and 3. above using Matlab and the following sequences:

(a) $x_1(n) = 3\sin(0.04\pi n)R_{200}(n)$:

```
% P5.36
%% P0536a,m
% x1(n)  = 3 sin(0.04¦Đn)R200(n):
clear;clc;close all;
N = 200; n = 0:N-1; x1 = 3*sin(0.04*pi*n); l = 4;
k = 0:N-1; X1 = imag(fft(x1,N));
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P5.36.4(a)');
subplot(2,1,1); H_s1 = stem(n,x1,'g','filled');
```

```
set(H_s1,'markersize',1);
title('Sequence: {\itx}_1({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-4,4]); xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X1,'r','filled');
set(H_s2,'markersize',2);
title('Imaginary-part of the DFT:
{\itX}_1_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-350,350]);
xlabel('\itk');
set(gca,'xtick',[0,l,N-l],'ytick',[-300,0,300])
print -deps2 ../epsfiles/P0536a
```

The sequence $x_1(n)$ and its DFT $X_1(k)$ are shown in Figure 5.30.

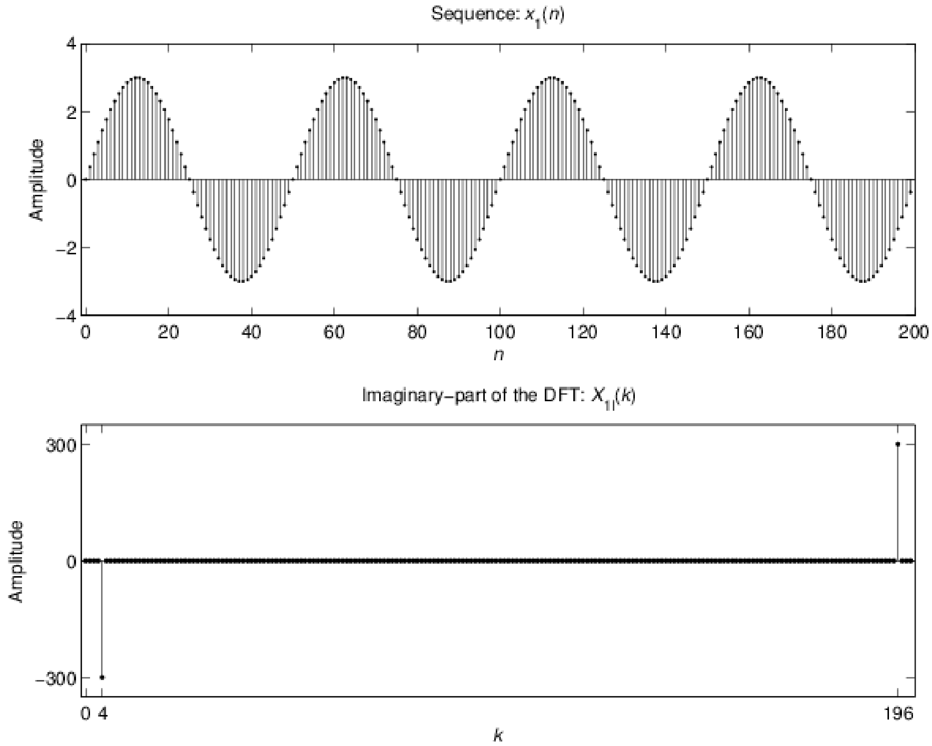


Figure 5.30: The signal $x_1(n)$ and its DFT $X_1(k)$ in Problem P5.36.6(a)

(b) $x_2(n) = 5\sin(10\pi n)\mathbf{R}_{50}(n)$:

```
%% P0536b.m
N = 50; n = 0:N-1; x2 = 5*sin(10*pi*n); l = 0;
k = 0:N-1; X2 = imag(fft(x2,N));
Hf_2 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_2,'NumberTitle','off','Name','P5.36.4(b)');
subplot(2,1,1); H_s1 = stem(n,x2,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}_2({\itn})','fontsize',10);
```

```
ylabel('Amplitude'); axis([-1,N,-1,1]); xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X2,'r','filled');
set(H_s2,'markersize',2);
title('Imaginary-part of the DFT:
{\itX}_2_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-1,1]); xlabel('\itk');
set(gca,'xtick',[0,N-1],'ytick',[-1,0,1])
print -deps2 ../epsfiles/P0536b
```

The sequence $x_2(n)$ and its DFT $X_2(k)$ are shown in Figure 5.31.

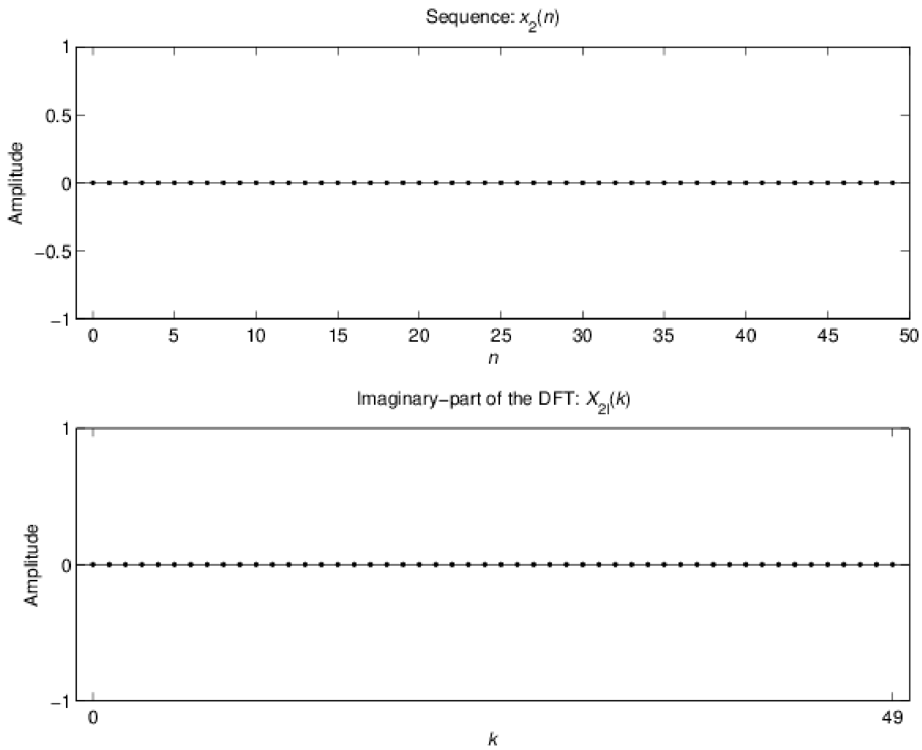


Figure 5.31: The signal $x_2(n)$ and its DFT $X_2(k)$ in Problem P5.34.6(b)

(c) $x_3(n) = [2\sin(0.5\pi n) + \sin(\pi n)]\mathbf{R}_{100}(n)$:

```
%% P0536c.m
N = 100; n = 0:N-1; x3 = 2*sin(0.5*pi*n)+0*sin(pi*n); l1
= 0; l2 = 25; l3 = 50
k = 0:N-1; X3 = imag(fft(x3,N));
Hf_3 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_3,'NumberTitle','off','Name','P5.36.4(c)');
subplot(2,1,1); H_s1 = stem(n,x3,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}_3({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-3,3]); xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X3,'r','filled');
```

```
set(H_s2,'markersize',2);
title('Imaginary-part of the DFT:
{\itX}_3_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-120,120]);
xlabel('\itk');
set(gca,'xtick',[l1,l2,l3,N-l2,N-1],'ytick',[-100,0,100])
print -deps2 ../epsfiles/P0536c
```

The sequence $x_3(n)$ and its DFT $X_3(k)$ are shown in Figure 5.32.

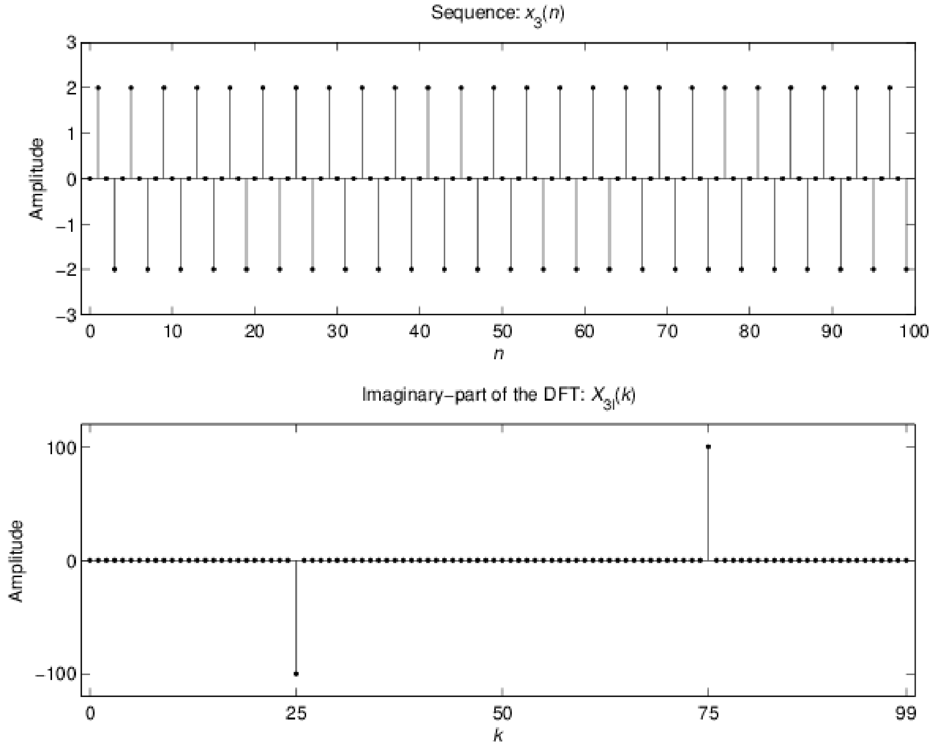


Figure 5.32: The signal $x_3(n)$ and its DFT $X_3(k)$ in Problem P5.34.6(c)

(d) $x_4(n) = \sin(25\pi n/16)\mathbf{R}_{64}(n)$:

```
%% P0536d.m
N = 64; n = 0:N-1; x4 = sin(25*pi*n/16); l = 50;
k = 0:N-1; X4 = imag(fft(x4,N));
Hf_4 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_4,'NumberTitle','off','Name','P5.36.4(d)');
subplot(2,1,1); H_s1 = stem(n,x4,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}_4({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-1.1,1.1]);
xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X4,'r','filled');
set(H_s2,'markersize',2);
title('Imaginary-part of the DFT:
```

```
{\itX}_4_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-40,40]); xlabel('\itk');
set(gca,'xtick',[0,N-1,1,N-1],'ytick',[-32,0,32])
print -deps2 ../epsfiles/P0536d
```

The sequence $x_4(n)$ and its DFT $X_4(k)$ are shown in Figure 5.33.

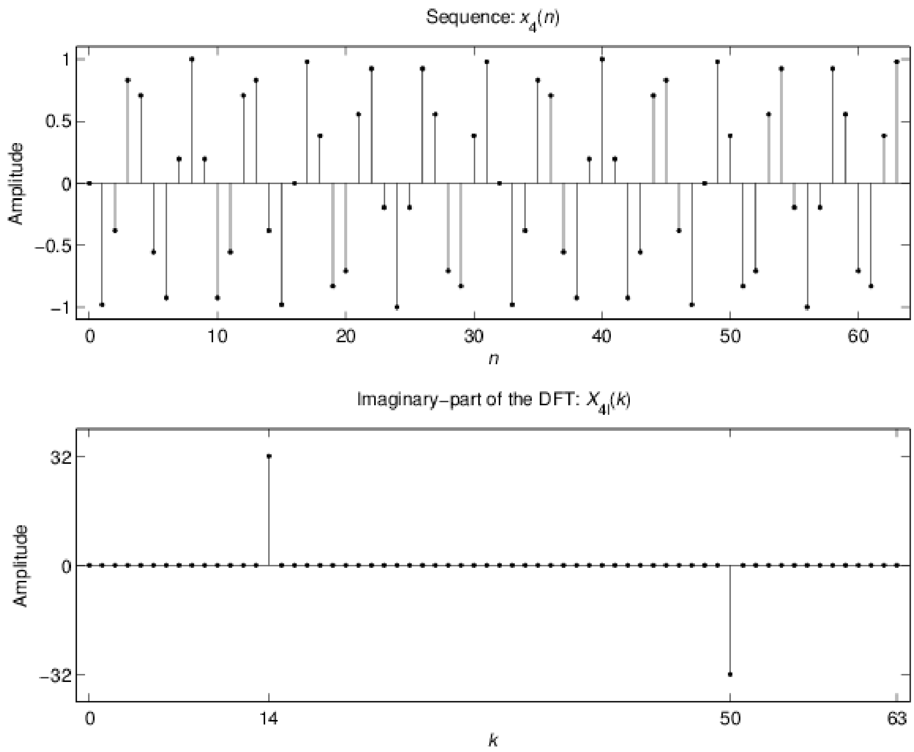


Figure 5.33: The signal $x_4(n)$ and its DFT $X_4(k)$ in Problem P5.34.6(d)

(e) $x_5(n) = [4\sin(0.1\pi n) - 3\sin(1.9\pi n)]\mathbf{R}_{20}(n)$:

```
%% P0536e.m
N = 20; n = 0:N-1; x5 = 4*sin(0.1*pi*n)-3*sin(1.9*pi*n);
l1 = 1; l2 = 19;
k = 0:N-1; X5 = imag(fft(x5,N));
Hf_5 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_5,'NumberTitle','off','Name','P5.36.4(e)');
subplot(2,1,1); H_s1 = stem(n,x5,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}_5({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-10,10]); xlabel('\itn');
subplot(2,1,2); H_s2 = stem(n,X5,'r','filled');
set(H_s2,'markersize',2);
title('Imaginary-part of the DFT:
{\itX}_5_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-80,80]); xlabel('\itk');
```

```
set(gca,'xtick',[0,l1,l2,N],'ytick',[-70,0,70])
print -deps2 ../epsfiles/P0536e
```

The sequence $x_5(n)$ and its DFT $X_5(k)$ are shown in Figure 5.34.

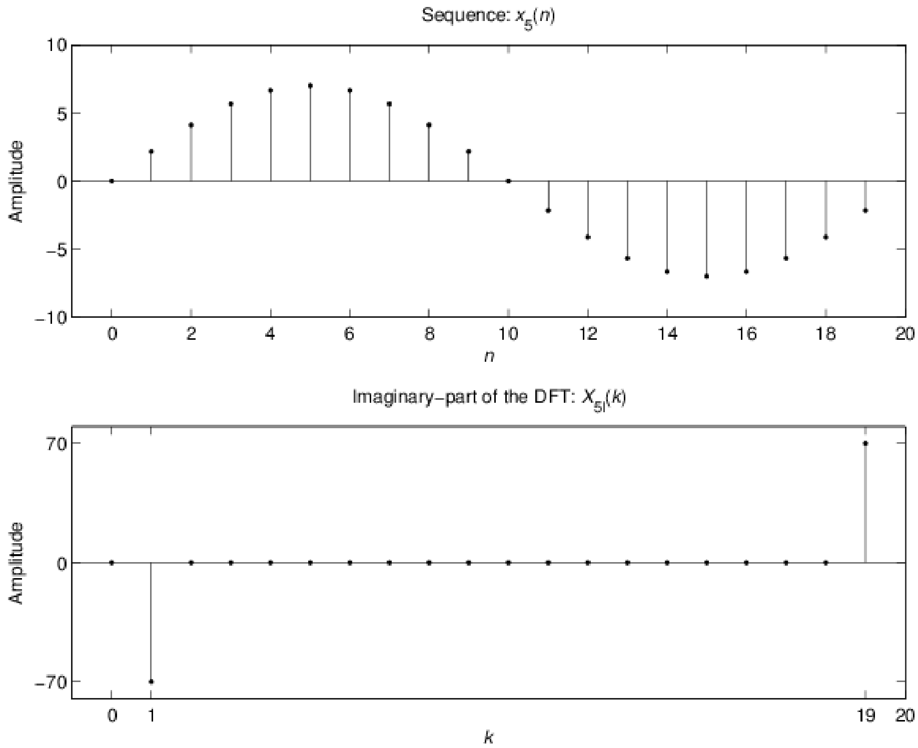


Figure 5.34: The signal $x_5(n)$ and its DFT $X_5(k)$ in Problem P5.34.6(e)

P5.37

Let $x(n) = A\sin(\omega_0 n)\ \mathbf{R}_N(n)$, where ω_0 is a real number.
1. Using the properties of the DFT, show that the real and the imaginary parts of $X(k)$ are given by

$$
\begin{aligned}
X(k) &= X_R(k) + jX_I(k) \\
X_R(k) &= -(A/2)\sin\left[\tfrac{\pi(N-1)}{N}(k-f_0N)\right]\frac{\sin[\pi(k-f_0N)]}{\sin[\pi(k-f_0N)/N]} \\
&\quad + (A/2)\sin\left[\tfrac{\pi(N-1)}{N}(k+f_0N)\right]\frac{\sin[\pi(k-N+f_0N)]}{\sin[\pi(k-N+f_0N)/N]} \\
X_I(k) &= -(A/2)\cos\left[\tfrac{\pi(N-1)}{N}(k-f_0N)\right]\frac{\sin[\pi(k-f_0N)]}{\sin[\pi(k-f_0N)/N]} \\
&\quad + (A/2)\cos\left[\tfrac{\pi(N-1)}{N}(k+f_0N)\right]\frac{\sin[\pi(k-N+f_0N)]}{\sin[\pi(k-N+f_0N)/N]}
\end{aligned}
$$

2. This result is the leakage property of sines. Explain it using the periodic extension $\tilde{x}(n)$ of $x(n)$ and the result in Problem P5.36.1.

3. Verify the leakage property using $x(n) = \sin(5\pi n/99)\ \ R_{100}(n)$. Plot the real and the imaginary parts of $X(k)$ using the **stem** function.

Solutions

Let $x(n) = A\sin(\omega_0 n)\mathbf{R}_{\boldsymbol{N}}(n)$, where ω_0 is a real number.

1. Consider

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j\left(\frac{2\pi}{N}\right)nk} = A\sum_{n=0}^{N-1}\sin(\omega_0 n)\left\{\cos\left(\frac{2\pi}{N}nk\right) - j\sin\left(\frac{2\pi}{N}nk\right)\right\}$$

$$X_{\mathrm{R}}(k) + jX_{\mathrm{I}}(k) = A\sum_{n=0}^{N-1}\sin(\omega_0 n)\cos\left(\frac{2\pi}{N}nk\right) - jA\sum_{n=0}^{N-1}\sin(\omega_0 n)\sin\left(\frac{2\pi}{N}nk\right)$$

Hence

$$X_{\mathrm{R}}(k) = A\sum_{n=0}^{N-1}\sin(\omega_0 n)\cos\left(\frac{2\pi}{N}nk\right) \tag{5.11}$$

$$X_{\mathrm{I}}(k) = -A\sum_{n=0}^{N-1}\sin(\omega_0 n)\sin\left(\frac{2\pi}{N}nk\right) \tag{5.12}$$

Consider the real-part in (5.11),

$$\begin{aligned}
X_{\mathrm{R}}(k) &= A\sum_{n=0}^{N-1}\sin(\omega_0 n)\cos\left(\frac{2\pi}{N}nk\right) = \frac{A}{2}\sum_{n=0}^{N-1}\left\{\sin\left(\omega_0 n - \frac{2\pi}{N}nk\right) + \sin\left(\omega_0 n + \frac{2\pi}{N}nk\right)\right\} \\
&= \frac{A}{2}\sum_{n=0}^{N-1}\left\{\sin\left(2\pi f_0 n - \frac{2\pi}{N}nk\right) + \sin\left(2\pi f_0 n + \frac{2\pi}{N}nk\right)\right\} \qquad [\because \omega_0 = 2\pi f_0] \\
&= \frac{A}{2}\sum_{n=0}^{N-1}\left\{\sin\left[\frac{2\pi}{N}(f_0 N - k)n\right] + \sin\left[\frac{2\pi}{N}(f_0 N + k)n\right]\right\} \\
&= \frac{A}{2}\sum_{n=0}^{N-1}\left\{-\sin\left[\frac{2\pi}{N}(k - f_0 N)n\right] + \sin\left[\frac{2\pi}{N}\{k - (N - f_0 N)\}n\right]\right\}, \quad 0 \le k < N
\end{aligned} \tag{5.13}$$

To reduce the sum-of-sine terms in (5.13), consider

$$\begin{aligned}
\sum_{n=0}^{N-1}\sin\left(\frac{2\pi}{N}vn\right) &= \frac{1}{j2}\sum_{n=0}^{N-1}e^{j\left(\frac{2\pi}{N}\right)vn} - \frac{1}{j2}\sum_{n=0}^{N-1}e^{-j\left(\frac{2\pi}{N}\right)vn} = \frac{1}{j2}\left(\frac{1-e^{j2\pi v}}{1-e^{j\frac{2\pi}{N}v}}\right) - \frac{1}{j2}\left(\frac{1-e^{-j2\pi v}}{1-e^{-j\frac{2\pi}{N}v}}\right) \\
&= \frac{1}{j2}e^{j\pi v\left(\frac{N-1}{N}\right)}\frac{\sin(\pi v)}{\sin(\pi v/N)} - \frac{1}{j2}e^{-j\pi n\left(\frac{N-1}{N}\right)}\frac{\sin(\pi v)}{\sin(\pi v/N)} \\
&= \sin\left\{\frac{\pi v(N-1)}{N}\right\}\frac{\sin(\pi v)}{\sin(\pi v/N)}
\end{aligned} \tag{5.14}$$

Now substituting (5.14) in the first term of (5.13) with v = $(k - f_0 N)$ and in the second term of (5.13) with v = $(k - [N - f_0 N])$, we obtain the desired result

$$X_R(k) = -\frac{A}{2}\sin\left\{\frac{\pi(N-1)}{N}(k-f_0N)\right\}\frac{\sin[\pi(f_0N-k)]}{\sin[\frac{\pi}{N}(f_0N-k)]} + \frac{A}{2}\sin\left\{\frac{\pi(N-1)}{N}(k-[N-f_0N])\right\}\frac{\sin\{\pi(k-[N-f_0N])\}}{\sin\{\frac{\pi}{N}(f_0N-k)\}} \tag{5.15}$$

Similarly, we can show that

$$X_I(k) = -\frac{A}{2}\sin\left\{\frac{\pi(N-1)}{N}(k-f_0N)\right\}\frac{\sin[\pi(f_0N-k)]}{\sin[\frac{\pi}{N}(f_0N-k)]} + \frac{A}{2}\sin\left\{\frac{\pi(N-1)}{N}(k-[N-f_0N])\right\}\frac{\sin\{\pi(k-[N-f_0N])\}}{\sin[\frac{\pi}{N}(f_0N-k)]} \tag{5.16}$$

2. The above result is the leakage property of sines. It implies that the original frequency ω_0 of the sine waveform has *leaked* into other frequencies that form the harmonics of the time-limited sequence. It is a natural result due to the fact that bandlimited periodic sines are sampled over noninteger periods. Due to this fact, the periodic extension of $x(n)$ does not result in a continuation of the sine waveform but has a jump at every N interval. This jump results in the leakage of one frequency into the abducent frequencies and hence the result of the Problem P5.36.1 do not apply.

3. Verification of the leakage property using $x(n) = \sin(5\pi n/99)\mathbf{R}_{100}(n)$:

```
% P5.37
% Matlab Verification
clear;clc;close all;
N = 100; n = 0:N-1; x = sin(5*pi*n/99); I = 5;
k = 0:N-1; X = fft(x,N);
Hf = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf,'NumberTitle','off','Name','P5.37.3');
subplot(3,1,1); H_s1 = stem(n,x,'g','filled');
set(H_s1,'markersize',2);
title('Sequence: {\itx}({\itn})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-1,1]); xlabel('\itn');
subplot(3,1,2); H_s2 = stem(n,real(X),'r','filled');
set(H_s2,'markersize',2);
title('Real-part of the DFT:
{\itX}_R({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-50,50]); xlabel('\itk');
set(gca,'xtick',[0,I,N/2,N-I,N],'ytick',[-50,0,50])
subplot(3,1,3); H_s3 = stem(n,imag(X),'g','filled');
set(H_s3,'markersize',2);
title('Imaginary-part of the DFT:
{\itX}_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-5,5]); xlabel('\itk');
set(gca,'xtick',[0,I,N/2,N-I,N],'ytick',[-5,0,5])
print -deps2 ../epsfiles/P0537
```

The sequence $x(n)$, the real-part of its DFT $X_R(k)$, and the imaginary part of its DFT $X_I(k)$ are shown in Figure 5.35.

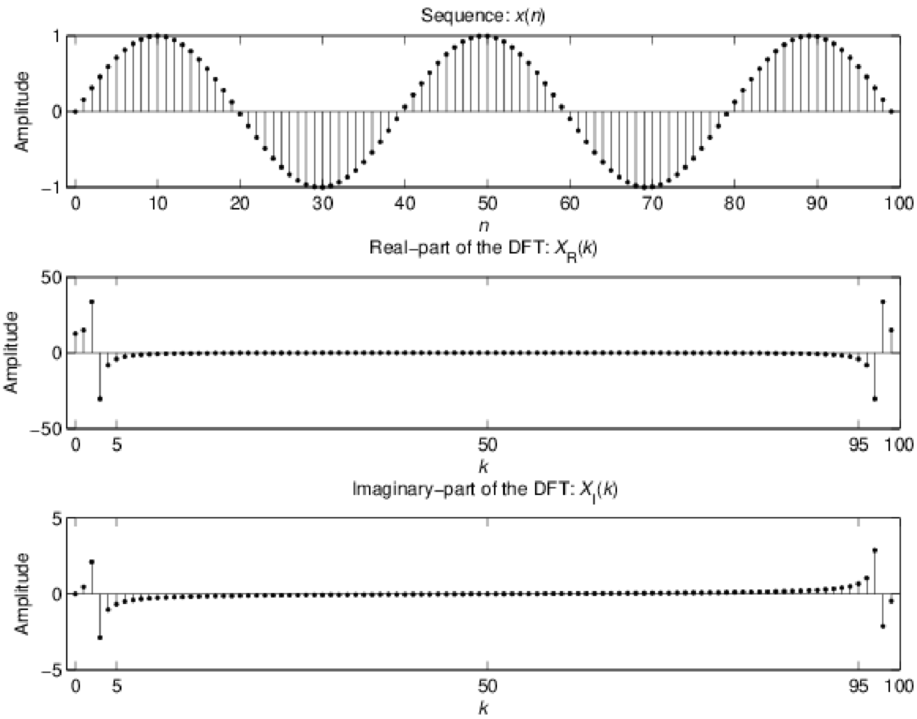


Figure 5.35: The leakage property of a sine signal in Problem P5.37.3

P5.38

An analog signal $x_a(t) = 2\sin(4\pi t) + 5\cos(8\pi t)$ is sampled at $t = 0.01n$ for $n = 0, 1, \ldots, N - 1$ to obtain an N-point sequence $x(n)$. An N-point DFT is used to obtain an estimate of the magnitude spectrum of $x_a(t)$.

1. From the following values of N, choose the one that will provide the accurate estimate of the spectrum of $x_a(t)$. Plot the real and imaginary parts of the DFT spectrum $X(k)$.
(a) $N = 40$, (b) $N = 50$, (c) $N = 60$.
2. From the following values of N, choose the one that will provide the least amount of leakage in the spectrum of $x_a(t)$. Plot the real and imaginary parts of the DFT spectrum $X(k)$.
(a) $N = 90$, (b) $N = 95$, (c) $N = 99$.

Solutions

1. Out of the given three values, $N = 50$ provides complete cycles of both the sine and the cosine components.

```
% P5.38
%% P0538a.m
N = 50; n =0:N-1; t = 0.01*n; I1 = 2; I2 =1;
x = 2*sin(4*pi*t)+5*cos(8*pi*t);
```

```
X = fft(x,N);
X_R = real(X);X_I = imag(X);
subplot(2,1,1); H_s1 = stem(n,real(X),'r','filled');
set(H_s1,'markersize',2);
title('Real-part of the DFT:
{\itX}_R({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-40,150]);
xlabel('\itk');
set(gca,'xtick',[0,I1,N/2,N-I1,N],'ytick',[-40,0,150])
subplot(2,1,2); H_s2 = stem(n,imag(X),'r','filled');
set(H_s2,'markersize',2);
title('Imaginary-part of the DFT:
{\itX}_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-60,60]); xlabel('\itk');
set(gca,'xtick',[0,I2,N/2,N-I2,N],'ytick',[-60,0,60])
print -deps2 ../epsfiles/P0538a
```

Thus $N = 50$ provides the most accurate estimate as shown in Figure 5.36.

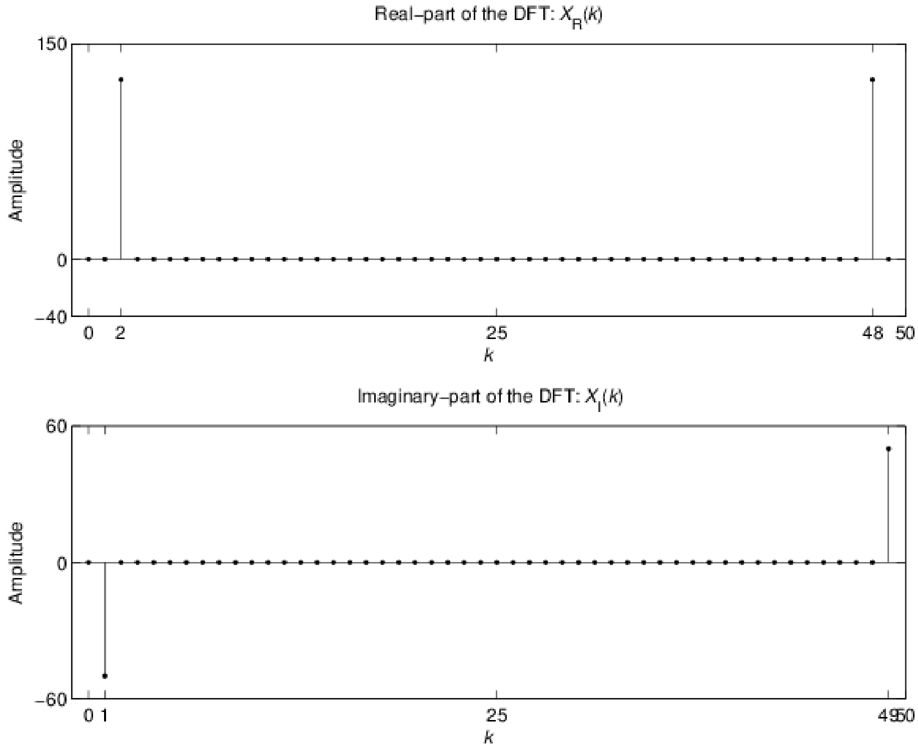


Figure 5.36: The accurate spectrum of the signal in Problem P5.38.1

2. Out of the given three values, $N = 99$ provides almost complete cycles of both the sine and the cosine components.

```
%% P0538b.m
N = 99; n =0:N-1; t = 0.01*n; I1 = 4; I2 =2;
x = 2*sin(4*pi*t)+5*cos(8*pi*t);
```

```
X = fft(x,N);
% X_R = real(X);X_I = imag(X);
subplot(2,1,1); H_s1 = stem(n,real(X),'r','filled');
set(H_s1,'markersize',2);
title('Real-part of the DFT:
{\itX}_R({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-30,300]);
xlabel('\itk');
set(gca,'xtick',[0,I1,N/2,N-I1,N],'ytick',[-30,0,300])
subplot(2,1,2); H_s2 = stem(n,imag(X),'r','filled');
set(H_s2,'markersize',2);
title('Imaginary-part of the DFT:
{\itX}_I({\itk})','fontsize',10);
ylabel('Amplitude'); axis([-1,N,-110,110]);
xlabel('\itk');
set(gca,'xtick',[0,I2,N/2,N-I2,N],'ytick',[-100,0,100])
print -deps2 ../epsfiles/P0538b
```

Thus $N = 99$ provides the least amount of leakage as shown in Figure 5.37.

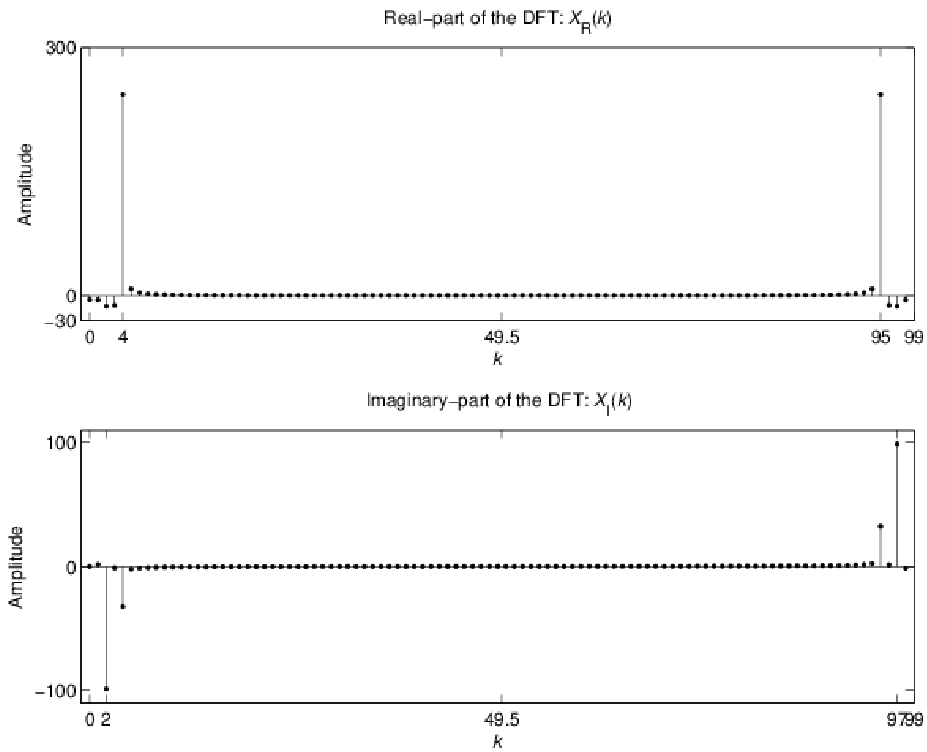


Figure 5.37: The least amount of leakage in the spectrum of the signal in Problem P5.38.2

P5.39

Using (5.49), determine and draw the signal flow graph for the $N = 8$ point, radix-2 decimation-in-frequency FFT algorithm. Using this flow graph, determine the DFT of the sequence

$$x(n) = \cos(\pi n/2), \quad 0 \le n \le 7$$

P5.40

Using (5.49), determine and draw the signal flow graph for the $N = 16$ point, radix-4 decimation-in-time FFT algorithm. Using this flow graph, determine the DFT of the sequence

$$x(n) = \cos(\pi n/2), \quad 0 \le n \le 15$$

P5.41

Let $x(n)$ be a uniformly distributed random number between [-1, 1] for $0 \le n \le 106$. Let

$$h(n) = \sin(0.4\pi n), \quad 0 \le n \le 100$$

1. Using the **conv** function, determine the output sequence $y(n) = x(n) * h(n)$.
2. Consider the overlap-and-save method of block convolution along with the FFT algorithm to implement high-speed block convolution. Using this approach, determine $y(n)$ with FFT sizes of 1024, 2048, and 4096.
3. Compare these approaches in terms of the convolution results and their execution times.

Solutions

3. K = 1024:

```
% P5.41
clear;
N = 10^6;
nx = 0:N-1;
x =2*rand([1,N])-1;
nh = 0:100;
h = sin(0.4*pi*nh);
t1 = clock;y1 = conv(h,x);t2 = clock;etime(t2,t1)
K = 1024;
% K = 2048;
% K = 4096;
t3 = clock;y2 = hsolpsav(x,h,K);t4 = clock;etime(t4,t3)
diff = max(abs([y1,zeros(1,length(y2)-length(y1))]-y2))

ans =
    0.0330
ans =
    0.0920
diff =
```

```
   2.1316e-14
```

K = 2048:

```
% P5.41
clear;
N = 10^6;
nx = 0:N-1;
x =2*rand([1,N])-1;
nh = 0:100;
h = sin(0.4*pi*nh);
t1 = clock;y1 = conv(h,x);t2 = clock;etime(t2,t1)
% K = 1024;
K = 2048;
% K = 4096;
t3 = clock;y2 = hsolpsav(x,h,K);t4 = clock;etime(t4,t3)
diff = max(abs([y1,zeros(1,length(y2)-length(y1))]-y2))

ans =
    0.0300
ans =
    0.0650
diff =
   1.9540e-14
```

K = 4096:

```
% P5.41
clear;
N = 10^6;
nx = 0:N-1;
x =2*rand([1,N])-1;
nh = 0:100;
h = sin(0.4*pi*nh);
t1 = clock;y1 = conv(h,x);t2 = clock;etime(t2,t1)
% K = 1024;
% K = 2048;
K = 4096;
t3 = clock;y2 = hsolpsav(x,h,K);t4 = clock;etime(t4,t3)
diff = max(abs([y1,zeros(1,length(y2)-length(y1))]-y2))

ans =
    0.0330
ans =
    0.0700
diff =
   1.7764e-14
```

Chapter 6

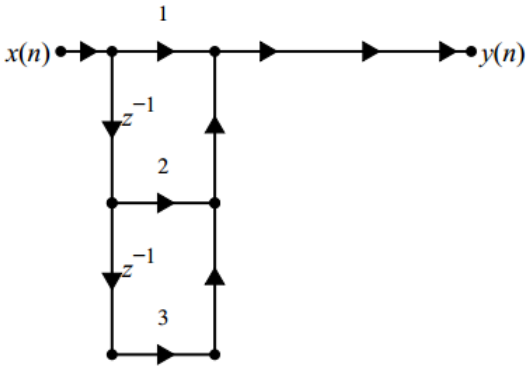
P6.1

Draw direct form I block diagram structures for each of the following LTI systems with input node $x(n)$ and output node $y(n)$.

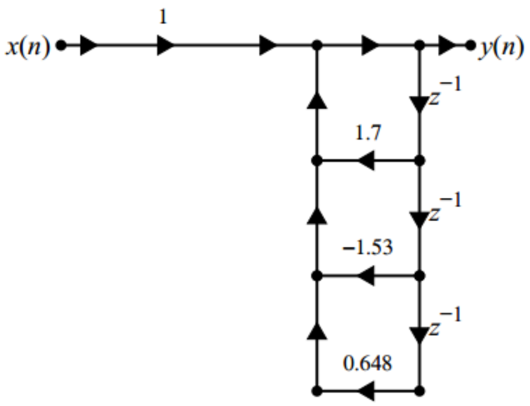
1. $y(n) = x(n) + 2\,x(n-1) + 3x(n-2)$
2. $H(z) = \dfrac{1}{1 - 1.7z^{-1} + 1.53z^{-2} - 0.648z^{-3}}$
3. $y(n) = 1.7\,y(n-1) - 1.36\,y(n-2) + 0.576\,y(n-3) + x(n)$
4. $y(n) = 1.6\,y(n-1) + 0.64\,y(n-2) + x(n) + 2\,x(n-1) + x(n-2)$
5. $H(z) = \dfrac{1 - 3z^{-1} + 3z^{-2} + z^{-3}}{1 + 0.2z^{-1} - 0.14z^{-2} + 0.44z^{-3}}$

Solutions

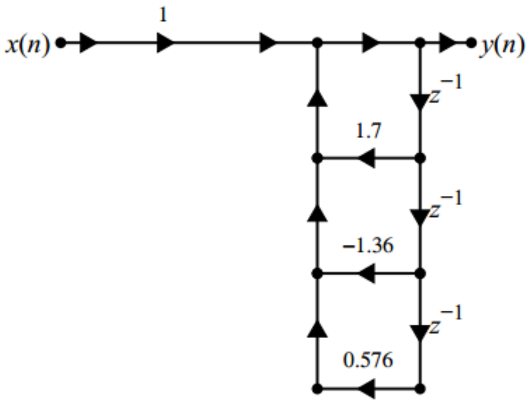
1. $y(n) = x(n) + 2x(n - 1) + 3x(n - 2)$:



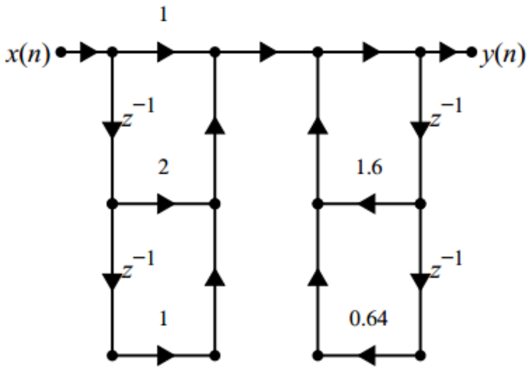
2. $H(z) = \dfrac{1}{1-1.7z^{-1}+1.53z^{-2}-0.648z^{-3}}$:



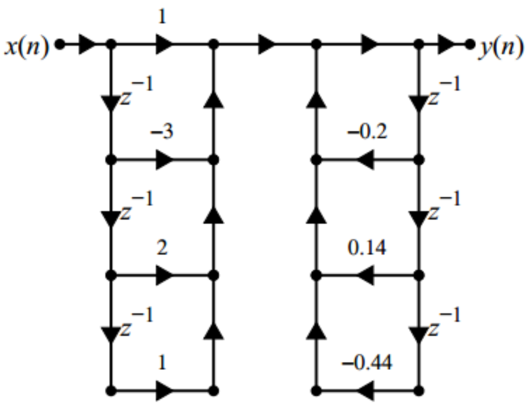
3. $y(n) = 1.7y(n - 1) - 1.36y(n - 2) + 0.576y(n - 3) + x(n)$:



4. $y(n) = 1.6y(n - 1) + 0.64y(n - 2) + x(n) + 2x(n - 1) + x(n - 2)$:



5. $H(z) = \dfrac{1 - 3z^{-1} + 3z^{-2} + z^{-3}}{1 + 0.2z^{-1} - 0.14z^{-2} + 0.44z^{-3}}$:



P6.2

Two block diagrams are shown in Figure P6.1. Answer the following for each structure.

1. Determine the system function $H(z) = Y(z)/X(z)$.

2. Is the structure canonical (i.e., with the least number of delays)? If not, draw a canonical structure.

3. Determine the value of K so that $H(e^{j0}) = 1$.

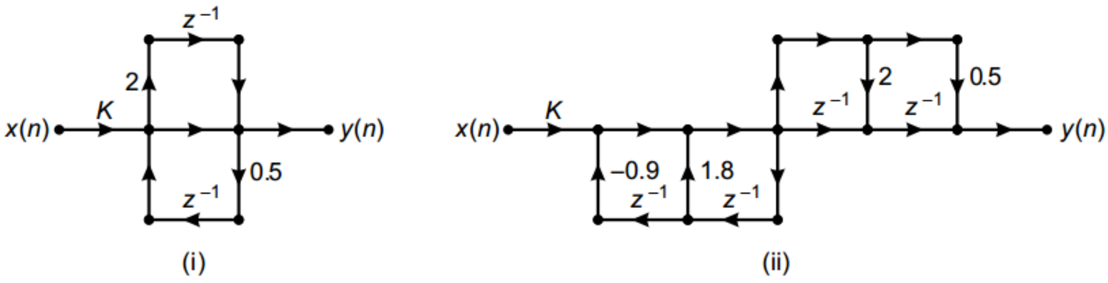


FIGURE P6.1 *Block diagrams for Problem 6.2*

Solutions

1. The system function $H(z) = Y(z)/\,X(z)$:

(i) Referring to signal nodes in the above figure (i):

$$\begin{aligned} w(n) &= K\,x(n) + \tfrac{1}{2}y(n-1) \\ y(n) &= w(n) + 2w(n-1) = K\,x(n) + \tfrac{1}{2}y(n-1) + 2K\,x(n-1) + 2\tfrac{1}{2}y(n-2) \\ &= K\,x(n) + 2K\,x(n-1) + \tfrac{1}{2}y(n-1) + y(n-2) \end{aligned}$$

Hence

$$H(z) = K\frac{1+2z^{-1}}{1-\frac{1}{2}z^{-1}-z^{-2}} \tag{6.1}$$

(ii) Referring to signal nodes in the above figure (ii):

$$\begin{aligned} w(n) &= K\,x(n) + 1.8w(n-1) - 0.9w(n-2) \Rightarrow W(z) = K\frac{X(z)}{1-0.8z^{-1}+0.2z^{-2}} \\ y(n) &= 0.5w(n) + 2w(n-1) + w(n-2) \Rightarrow Y(z) = \left(0.5 + 2z^{-1} + z^{-2}\right)W(z) \end{aligned}$$

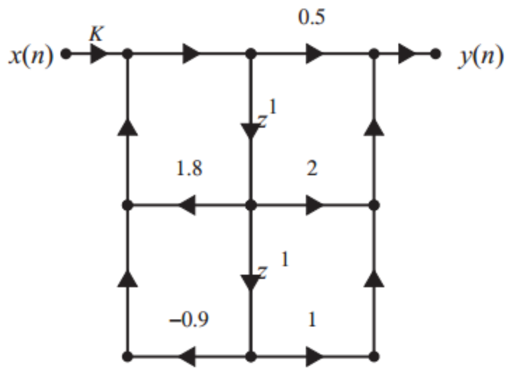
Hence

$$H(z) = \frac{Y(z)}{X(z)} = K\frac{0.5+2z^{-1}+z^{-2}}{1-0.8z^{-1}+0.2z^{-2}} \tag{6.2}$$

2. Canonical structure:

(i) The given structure is canonical

(ii) The given structure is not canonical. The canonical structure is



3. Value of K so that $H(e^{j0}) = 1$:

(i) From (6.1), $H(1) = 1 = K\dfrac{1+2}{1-\dfrac{1}{2}-1} = -6K \Rightarrow K = -\dfrac{1}{6}$.

(ii) From (6.2), $H(1) = 1 = K\dfrac{0.5+2+1}{1-1.8+0.9} = 35K \Rightarrow K = -\dfrac{1}{35}$

P6.3

Consider the LTI system described by

$$y(n) = a\,y(n-1) + b\,x(n) \tag{6.82}$$

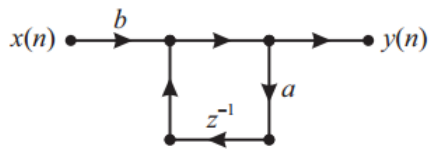
1. Draw a block diagram of this system with input node $x(n)$ and output node $y(n)$.
2. Now perform the following two operations on the structure drawn in part 1: (i) reverse all arrow directions and (ii) interchange the input node with the output node. Notice that the branch node becomes the adder node and vice versa. Redraw the block diagram so that input node is on the left side and the output node is on the right side. This is the *transposed* block diagram.
3. Determine the difference equation representation of your transposed structure in part 2, and verify that it is the same equation as (6.82).

Solutions

Consider the LTI system described by

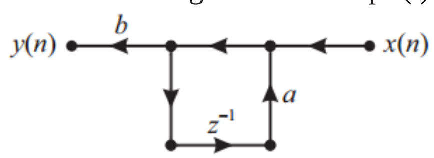
$$y(n) = a\,y(n-1) + b\,x(n) \tag{6.3}$$

1. Block diagram of the above system with input node $x(n)$ and output node $y(n)$ is shown below.



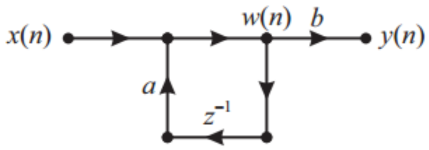
(a)

2. *Transposed* block diagram: The block diagram due to steps (i) and (ii) is shown below.



(i) and (ii)

The final block diagram after redrawing is shown below.



(b)

3. Difference equation representation of the transposed structure in part 2 above: Referring to the nodes in the transposed structure above

$$w(n) = x(n) + a\,w(n-1) \Rightarrow W(z) = \frac{1}{1 - az^{-1}} X(z)$$

and

$$w(n) = x(n) + a\,w(n-1) \Rightarrow W(z) = \frac{1}{1 - az^{-1}} X(z)$$

or

$$y(n) = b\,w(n) \Rightarrow Y(z) = b\,W(z)$$

which is the same equation as (6.3).

P6.4

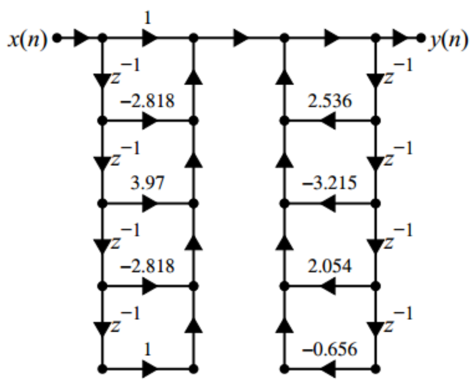
Consider the LTI system given by

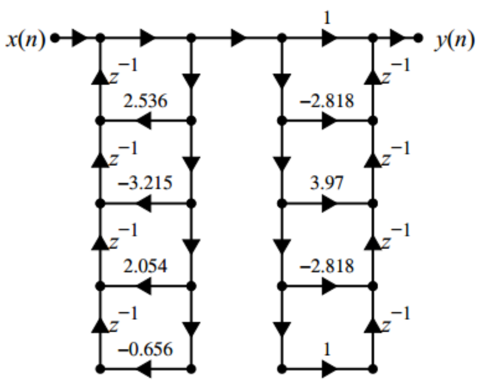
$$H(z) = \frac{1 - 2.818z^{-1} + 3.97z^{-2} - 2.8180z^{-3} + z^{-4}}{1 - 2.536z^{-1} + 3.215z^{-2} - 2.054z^{-3} + 0.6560z^{-4}} \tag{6.83}$$

1. Draw the normal direct form I structure block diagram.
2. Draw the transposed direct form I structure block diagram.
3. Draw the normal direct form II structure block diagram. Observe that it looks very similar to that in part 2.
4. Draw the transposed direct form II structure block diagram. Observe that it looks very similar to that in part 1.

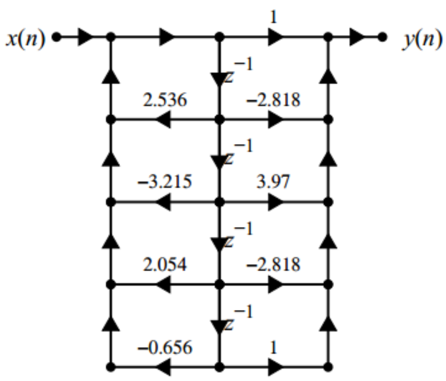
Solutions

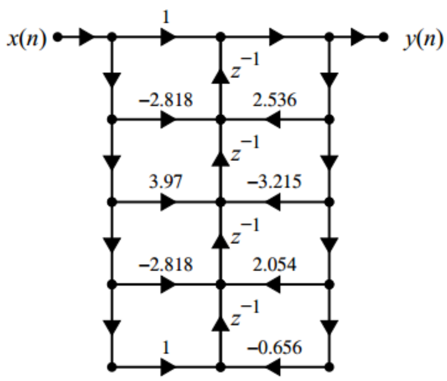
1. The normal direct form I structure block diagram is shown below on the left.





2. The transposed direct form I structure block diagram is shown above on the right.
3. The normal direct form II structure block diagram is shown below on the left.





Clearly it looks similar to that in part 2.
4. The transposed direct form II structure block diagram is shown above on the right. Clearly it looks similar to that in part 1.

P6.5

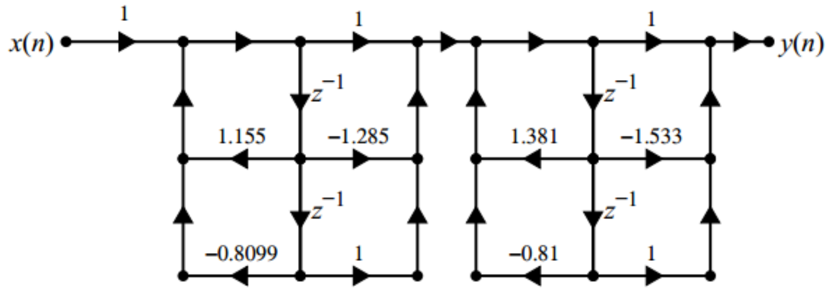
Consider the LTI system given in Problem P6.4.
1. Draw a cascade structure containing 2nd-order normal direct-form-II sections.
2. Draw a cascade structure containing 2nd-order transposed direct-form-II sections.
3. Draw a parallel structure containing 2nd-order normal direct-form-II sections.
4. Draw a parallel structure containing 2nd-order transposed direct-form-II sections.

Solutions

1. A cascade structure containing second-order normal direct from II sections: Matlab script:

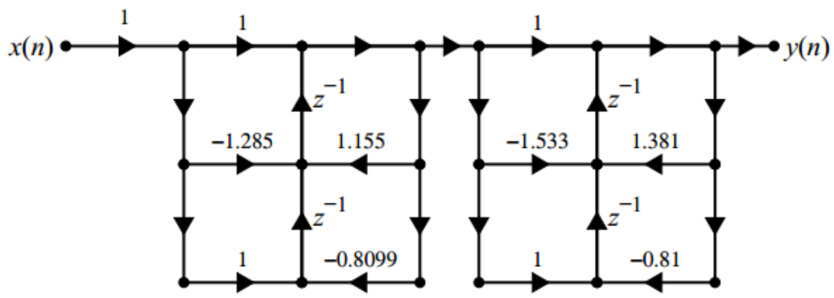
```
% P6.5
%% P0605a
b = [1,-2.818,3.97,-2.818,1]; a = [1,-2.536,3.215,-
2.054,0.656];
[b0,B,A] = dir2cas(b,a)
```

```
b0 =
     1
B =
    1.0000   -1.2854    1.0000
    1.0000   -1.5326    1.0000
A =
    1.0000   -1.1553    0.8099
    1.0000   -1.3807    0.8100
```



2. A cascade structure containing second-order transposed direct from II sections: Matlab script:

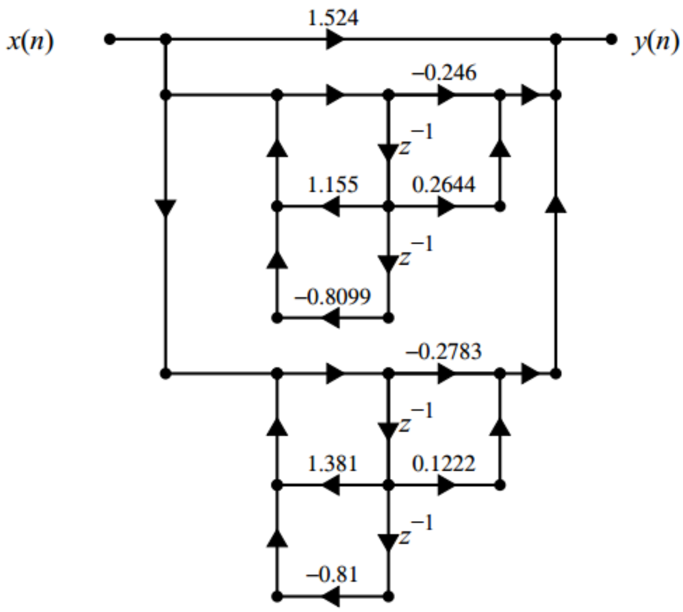
```
%% P0605b
b = [1,-2.818,3.97,-2.818,1]; a = [1,-2.536,3.215,-
2.054,0.656];
[b0,B,A] = dir2cas(b,a)

b0 =
     1
B =
    1.0000   -1.2854    1.0000
    1.0000   -1.5326    1.0000
A =
    1.0000   -1.1553    0.8099
    1.0000   -1.3807    0.8100
```



3. A parallel structure containing second-order normal direct from II sections: Matlab script:

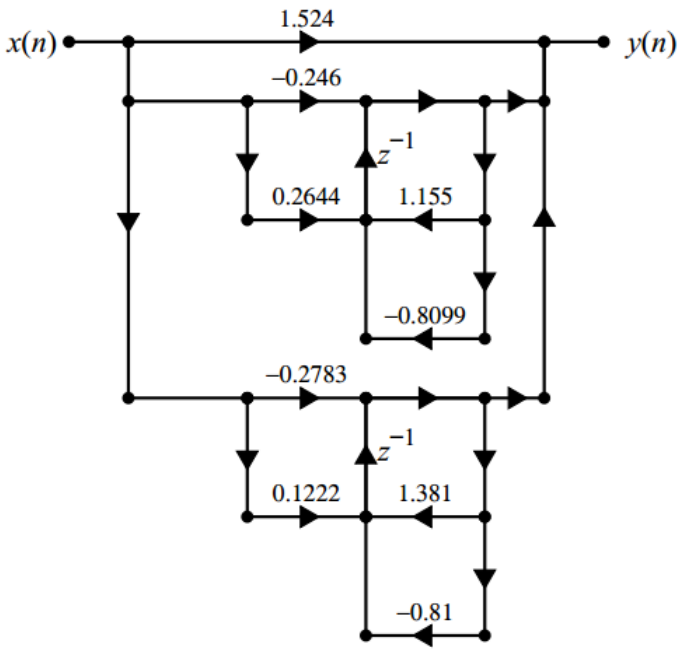
```
%% P0605c
b = [1,-2.818,3.97,-2.818,1];
a = [1,-2.536,3.215,-2.054,0.656];
[C,B,A] = dir2par(b,a)
```

```
C =
    1.5244
B =
   -0.2460    0.2644
   -0.2783    0.1222
A =
    1.0000   -1.1553    0.8099
    1.0000   -1.3807    0.8100
```



4. A parallel structure containing second-order transposed direct from II sections: Matlab script:

```
%% P0605d
b = [1,-2.818,3.97,-2.818,1];
a = [1,-2.536,3.215,-2.054,0.656];
[C,B,A] = dir2par(b,a)

C =
    1.5244
B =
   -0.2460    0.2644
   -0.2783    0.1222
A =
    1.0000   -1.1553    0.8099
    1.0000   -1.3807    0.8100
```



P6.6

A causal linear time-invariant system is described by

$$y(n)=\sum_{k=0}^{4}\cos(0.1\pi k)x(n-k)-\sum_{k=1}^{5}(0.8)^{k}\sin(0.1\pi k)y(n-k)$$

Determine and draw the block diagrams of the following structures. Compute the response of the system to

$$x(n)=[1+2(-1)^{n}],\quad 0\le n\le 50$$

in each case, using the following structures.

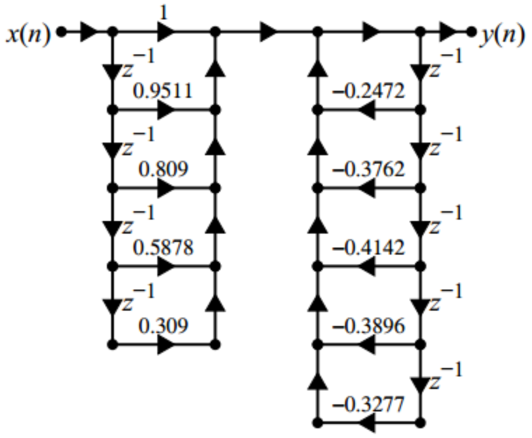
1. Normal direct form I
2. Transposed direct form II
3. Cascade form containing 2nd-order normal direct-form-II sections
4. Parallel form containing 2nd-order transposed direct-form-II sections
5. Lattice-ladder form

Solutions

1. Normal direct form I: Matlab script:

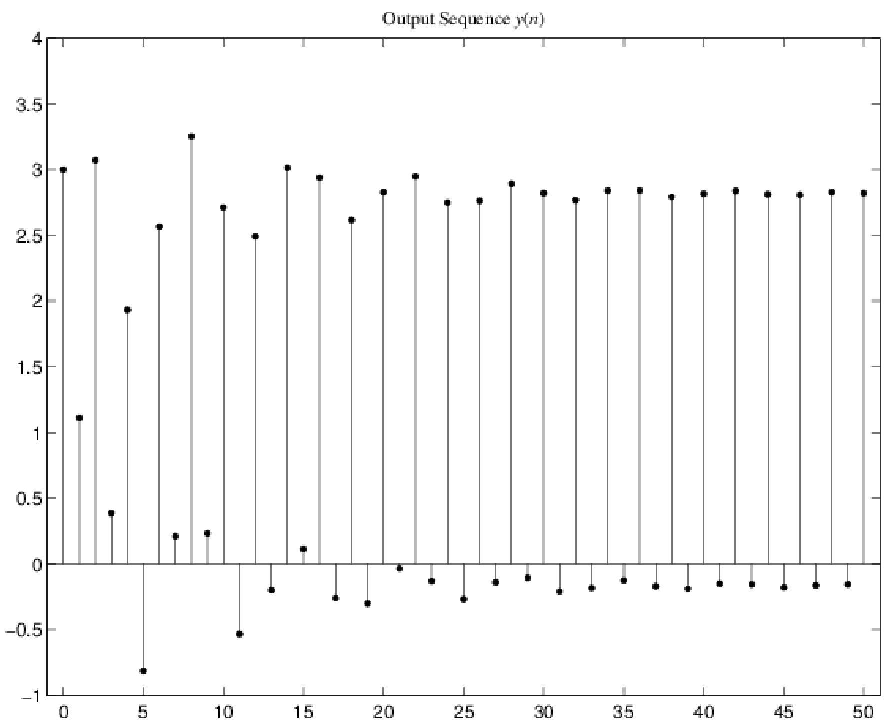
```
% P6.6
%% P0606a.m
% Normal direct form I
% clear;
b = cos(0.1*pi*[0:4]); a =
```

```
[1,((0.8).^[1:5]).*sin(0.1*pi*[1:5])];
n = 0:50; x = 1 + 2*(-1).^n; y1 = filter(b,a,x);
H_stem = stem(n,y1,'g','filled');
set(H_stem,'markersize',3); axis([-1,51,-1,4]);
title('Output Sequence
{\ity}({\itn})','fontname','times','fontweight','normal')
print -deps2 ../EPSFILES/P0606a
```



Response of the system to

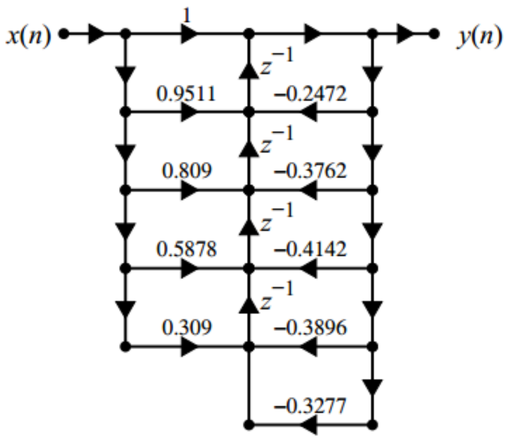
$$x(n) = \left[1 + 2(-1)^n\right], \quad 0 \le n \le 50$$



2. Transposed direct form II: Matlab script:

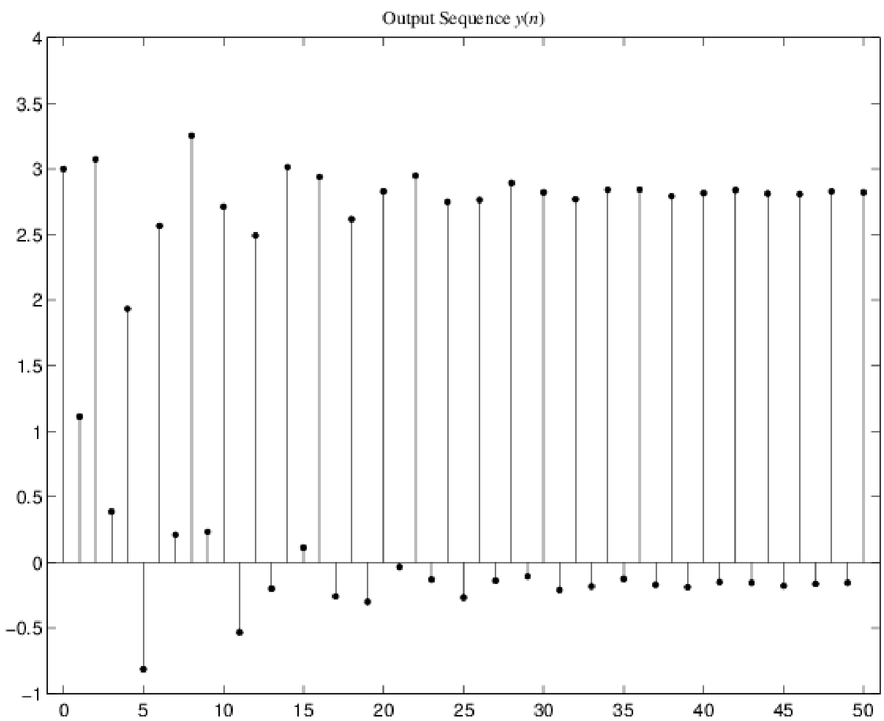
```
%% P0606b.m
% Transposed direct form II
```

```
% clear;
b = cos(0.1*pi*[0:4]); a =
[1,((0.8).^[1:5]).*sin(0.1*pi*[1:5])];
n = 0:50; x = 1 + 2*(-1).^n; y2 = filter(b,a,x);
H_stem = stem(n,y2,'g','filled');
set(H_stem,'markersize',3); axis([-1,51,-1,4]);
title('Output Sequence
{\ity}({\itn})','fontname','times','fontweight','normal')
print -deps2 ../EPSFILES/P0606b
```



Response of the system to

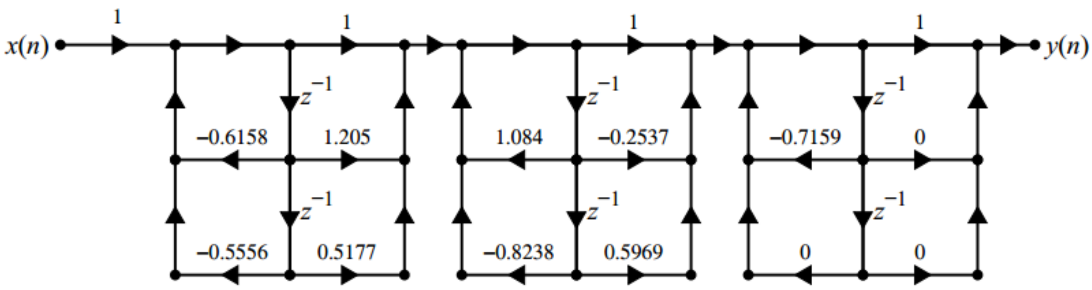
$$x(n) = \left[1 + 2(-1)^n\right], \quad 0 \le n \le 50$$



3. Cascade form containing second-order normal direct form II sections: Matlab script:

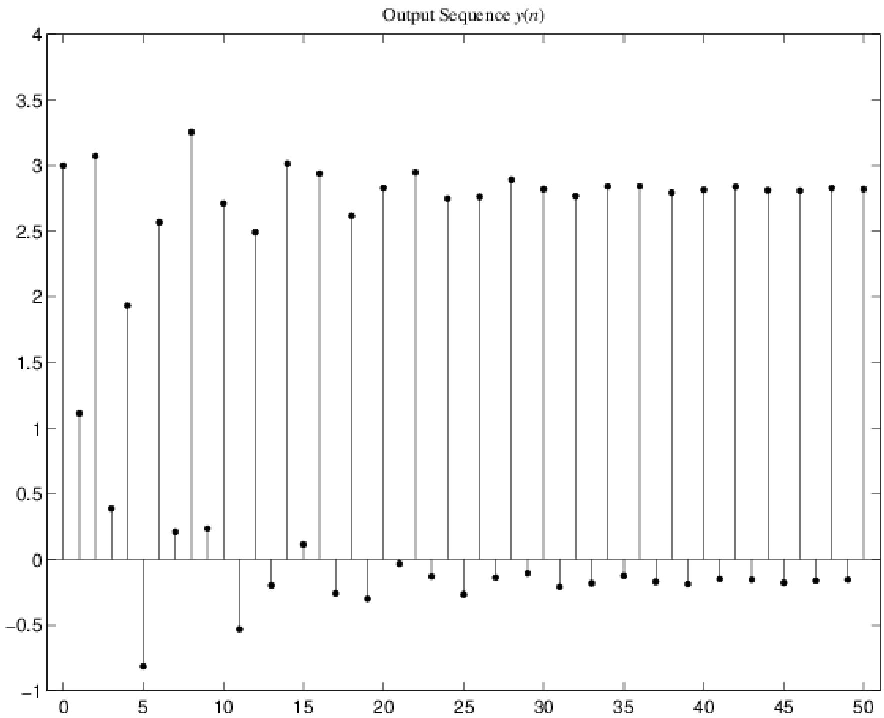
```
%% P0606c.m
```

```
% Cascade form containing second-order normal direct form
II sections
% clear;
b = cos(0.1*pi*[0:4]); a =
[1,((0.8).^[1:5]).*sin(0.1*pi*[1:5])];
[b0,B,A] = dir2cas(b,a)
n = 0:50; x = 1 + 2*(-1).^n; y3 = casfiltr(b0,B,A,x);
H_stem = stem(n,y3,'g','filled');
set(H_stem,'markersize',3); axis([-1,51,-1,4]);
title('Output Sequence
{\ity}({\itn})','fontname','times','fontweight','normal')
print -deps2 ../EPSFILES/P0606c

b0 =
     1
B =
    1.0000    1.2047    0.5177
    1.0000   -0.2537    0.5969
    1.0000         0         0
A =
    1.0000    0.6158    0.5556
    1.0000   -1.0844    0.8238
    1.0000    0.7159         0
```



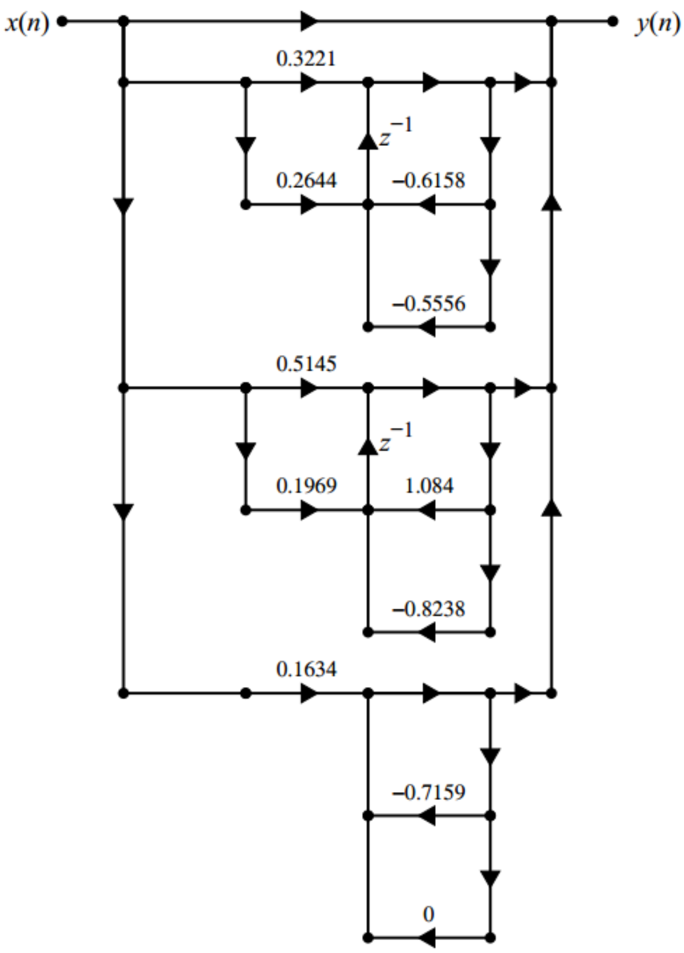
Response of the system to

$$x(n) = \left[1 + 2(-1)^n\right], \quad 0 \le n \le 50$$



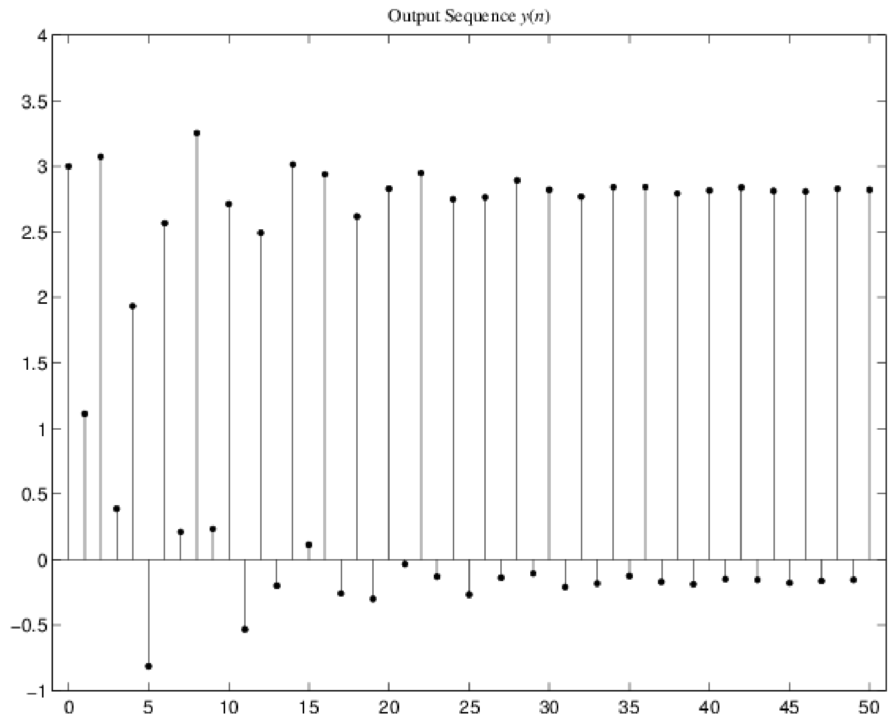
4. Parallel form containing second-order transposed direct form II sections: Matlab script:

```
%% P0606d.m
% Parallel form containing second-order transposed direct
form II sections
% clear;
b = cos(0.1*pi*[0:4]);
a = [1,((0.8).^[1:5]).*sin(0.1*pi*[1:5])];
[C,B,A] = dir2par(b,a)
n = 0:50; x = 1 + 2*(-1).^n; y4 = parfiltr(C,B,A,x);
H_stem = stem(n,y4,'g','filled');
set(H_stem,'markersize',3); axis([-1,51,-1,4]);
title('Output Sequence
{\ity}({\itn})','fontname','times','fontweight','normal')
print -deps2 ../EPSFILES/P0606d

C =

     []
B =
    0.3221    0.2644
    0.5145    0.1969
    0.1634         0
A =
    1.0000    0.6158    0.5556
    1.0000   -1.0844    0.8238
    1.0000    0.7159         0
```



Response of the system to

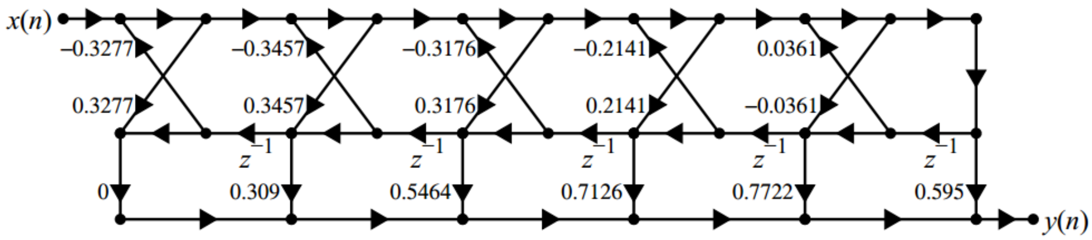
$$x(n) = [1 + 2(-1)^n], \quad 0 \le n \le 50$$



5. Lattice-ladder form: Matlab script:

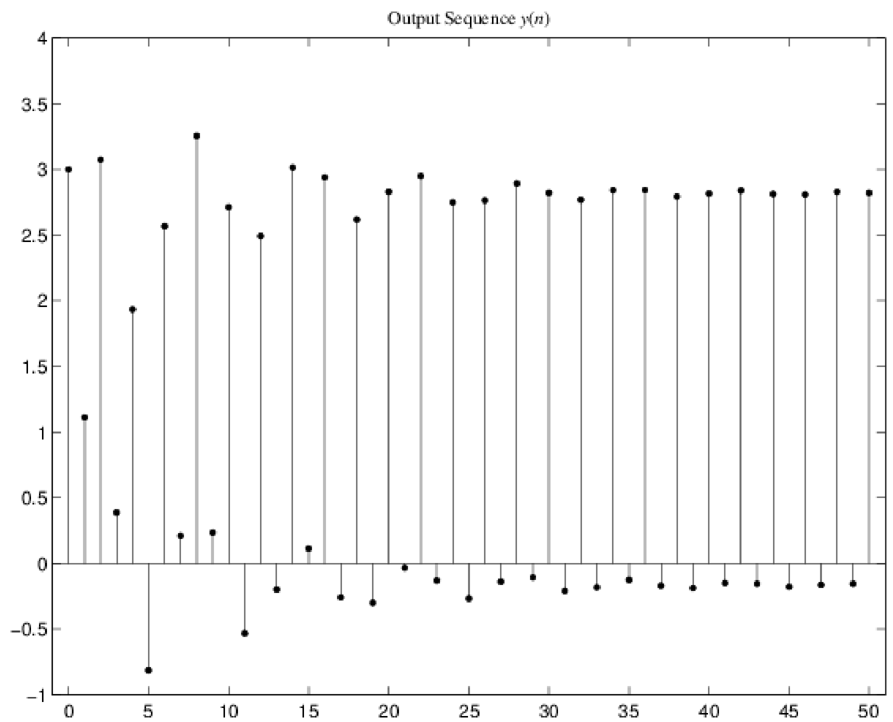
```
%% P0606e.m
% Lattice-ladder form
% clear;
b = cos(0.1*pi*[0:4]); a =
[1,((0.8).^[1:5]).*sin(0.1*pi*[1:5])];
[K,C] = dir2ladr(b,a)
n = 0:50; x = 1 + 2*(-1).^n; [y5] = ladrfilt(K,C,x);
H_stem = stem(n,y5,'g','filled');
set(H_stem,'markersize',3); axis([-1,51,-1,4]);
title('Output Sequence
{\ity}({\itn})','fontname','times','fontweight','normal')
print -deps2 ../EPSFILES/P0606e
```

```
K =
   -0.0361    0.2141    0.3176    0.3457    0.3277
C =
    0.5950    0.7722    0.7126    0.5464    0.3090         0
```



Response of the system to

$$x(n) = \left[1 + 2(-1)^n\right], \quad 0 \le n \le 50$$



P6.7

An IIR filter is described by the following system function

$$H(z) = 2\left(\frac{1+0z^{-1}+z^{-2}}{1-0.8z^{-1}+0.64z^{-2}}\right) + \left(\frac{2-z^{-1}}{1-0.75z^{-1}}\right) + \left(\frac{1+2z^{-1}+z^{-2}}{1+0.81z^{-2}}\right)$$

Determine and draw the following structures.

1. Transposed direct form I
2. Normal direct form II
3. Cascade form containing transposed 2nd-order direct-form-II sections
4. Parallel form containing normal 2nd-order direct-form-II sections
5. Lattice-ladder form

Solutions

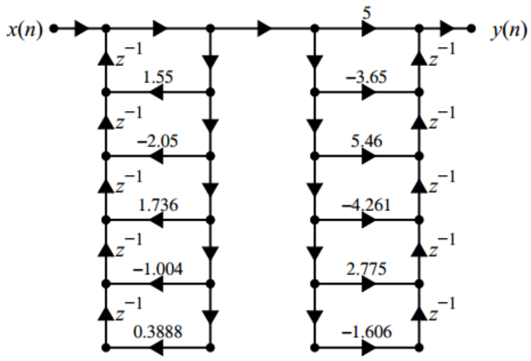
1. Transposed direct form I: Matlab script:

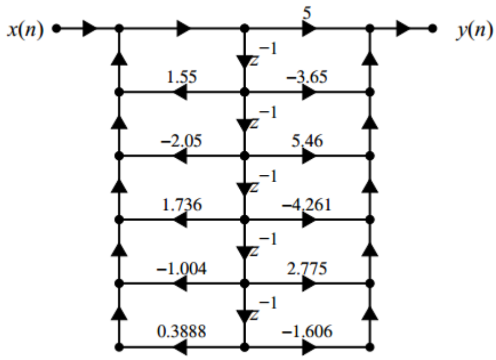
```
% P6.7
%% P0607a.m
% (a) Transposed Direct form-I
% Given H(z)
clear;
b1 = [2,0,1]; a1 = [1,-0.8,0.64]; [R1,p1,k1] =
residuez(b1,a1);
b2 = [2,-1]; a2 = [1,-0.75]; [R2,p2,k2] =
residuez(b2,a2);
b3 = [1,2,1]; a3 = [1,0,0.81]; [R3,p3,k3] =
residuez(b3,a3);
R = [R1;R2;R3]; p = [p1;p2;p3]; k = k1+k2+k3; [b,a] =
residuez(R,p,k)

b =
    5.0000   -3.6500    6.4600   -5.0110    3.5848   -2.2134
a =
    1.0000   -1.5500    2.0500   -1.7355    1.0044   -0.3888
```

2. Normal direct form II: Matlab script:

```
%% P0607b.m
% (b) Normal direct form-II
% Given H(z)
clear;
b1 = [2,0,2]; a1 = [1,-0.8,0.64]; [R1,p1,k1] =
residuez(b1,a1);
b2 = [2,-1]; a2 = [1,-0.75]; [R2,p2,k2] =
```

```
residuez(b2,a2);
b3 = [1,2,1]; a3 = [1,0,0.81]; [R3,p3,k3] =
residuez(b3,a3);
R = [R1;R2;R3]; p = [p1;p2;p3]; k = k1+k2+k3; [b,a] =
residuez(R,p,k)

b =
   5.0000   -3.6500    6.4600   -5.0110    3.5848   -2.2134
a =
   1.0000   -1.5500    2.0500   -1.7355    1.0044   -0.3888
```

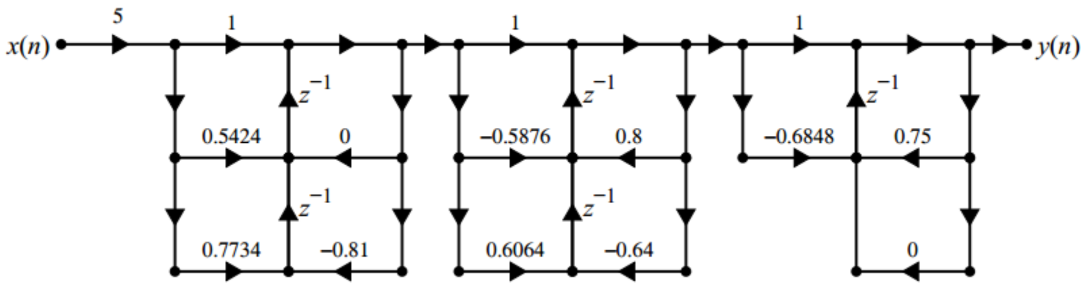




3. Cascade form containing transposed second-order direct form II sections: Matlab script:

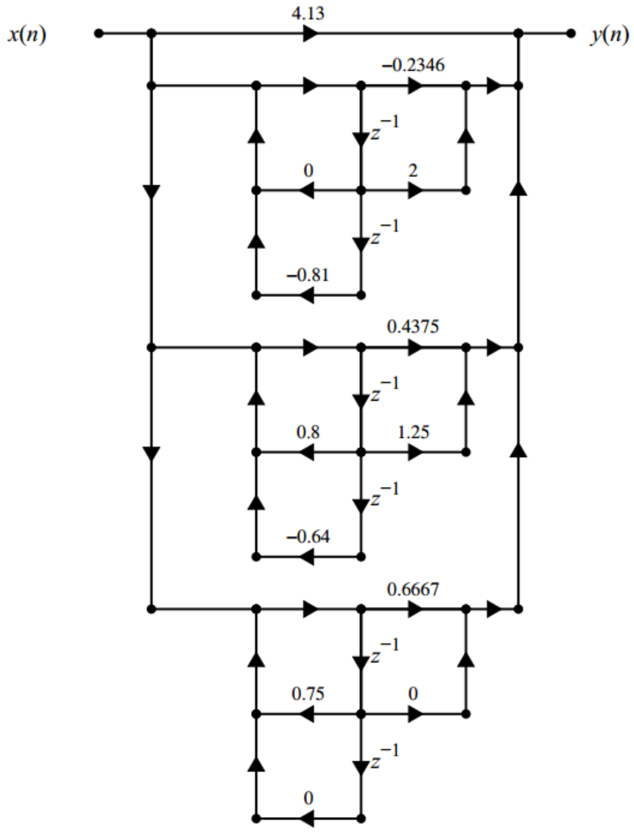
```
%% P0607c.m
% (c) Transposed Cascade form
% Given H(z)
clear;
b1 = [2,0,1]; a1 = [1,-0.8,0.64]; [R1,p1,k1] =
residuez(b1,a1);
b2 = [2,-1]; a2 = [1,-0.75]; [R2,p2,k2] =
residuez(b2,a2);
b3 = [1,2,1]; a3 = [1,0,0.81]; [R3,p3,k3] =
residuez(b3,a3);
R = [R1;R2;R3]; p = [p1;p2;p3]; k = k1+k2+k3; [b,a] =
residuez(R,p,k)
[b0,B,A] = dir2cas(b,a)

b =
   5.0000   -3.6500    5.4600   -4.2610    2.7748   -1.6059
a =
   1.0000   -1.5500    2.0500   -1.7355    1.0044   -0.3888
b0 =
     5
B =
   1.0000    0.5424    0.7734
   1.0000   -0.5876    0.6064
   1.0000   -0.6848         0
```

```
A =
    1.0000   -0.0000    0.8100
    1.0000   -0.8000    0.6400
    1.0000   -0.7500         0
```



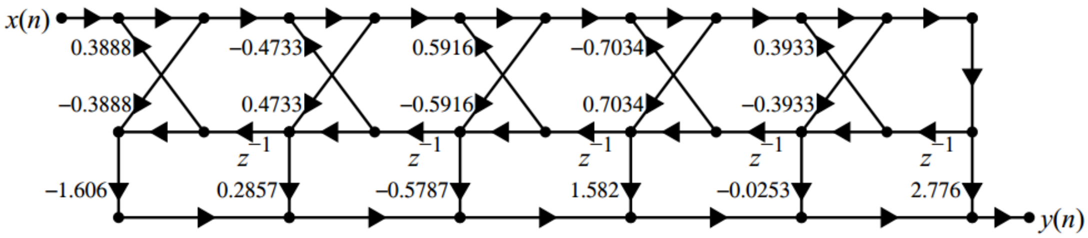
4. Parallel form containing normal second-order direct form II sections: Matlab script

```
%% P0607d.m
% (d) Normal Parallel form
% Given H(z)
clear;
b1 = [2,0,1]; a1 = [1,-0.8,0.64]; [R1,p1,k1] =
residuez(b1,a1);
b2 = [2,-1]; a2 = [1,-0.75]; [R2,p2,k2] =
residuez(b2,a2);
b3 = [1,2,1]; a3 = [1,0,0.81]; [R3,p3,k3] =
residuez(b3,a3);
R = [R1;R2;R3]; p = [p1;p2;p3]; k = k1+k2+k3; [b,a] =
residuez(R,p,k)
[C,B,A] = dir2par(b,a)

b =
    5.0000   -3.6500    5.4600   -4.2610    2.7748   -1.6059
a =
    1.0000   -1.5500    2.0500   -1.7355    1.0044   -0.3888
C =
    4.1304
B =
   -0.2346    2.0000
    0.4375    1.2500
    0.6667         0
A =
    1.0000   -0.0000    0.8100
    1.0000   -0.8000    0.6400
    1.0000   -0.7500         0
```



5. Lattice-ladder form: Matlab script:

```
%% P0607e.m
% Lattice-Ladder form
% Given H(z)
clear;
b1 = [2,0,1]; a1 = [1,-0.8,0.64]; [R1,p1,k1] =
residuez(b1,a1);
b2 = [2,-1]; a2 = [1,-0.75]; [R2,p2,k2] =
residuez(b2,a2);
b3 = [1,2,1]; a3 = [1,0,0.81]; [R3,p3,k3] =
residuez(b3,a3);
R = [R1;R2;R3]; p = [p1;p2;p3]; k = k1+k2+k3; [b,a] =
residuez(R,p,k)
[K,C] = dir2ladr(b,a)

b =
    5.0000   -3.6500    5.4600   -4.2610    2.7748   -1.6059
a =
    1.0000   -1.5500    2.0500   -1.7355    1.0044   -0.3888
K =
   -0.3933    0.7034   -0.5916    0.4733   -0.3888
C =
    2.7756   -0.0253    1.5817   -0.5787    0.2857   -1.6059
```



P6.8

An IIR filter is described by the following system function

$$H(z) = \left(\frac{-14.75 - 12.9z^{-1}}{1 - \frac{7}{8}z^{-1} + \frac{3}{32}z^{-2}}\right) + \left(\frac{24.5 + 26.82z^{-1}}{1 - z^{-1} + \frac{1}{2}z^{-2}}\right)\left(\frac{1 + 2z^{-1} + z^{-2}}{1 + 0.81z^{-2}}\right)$$

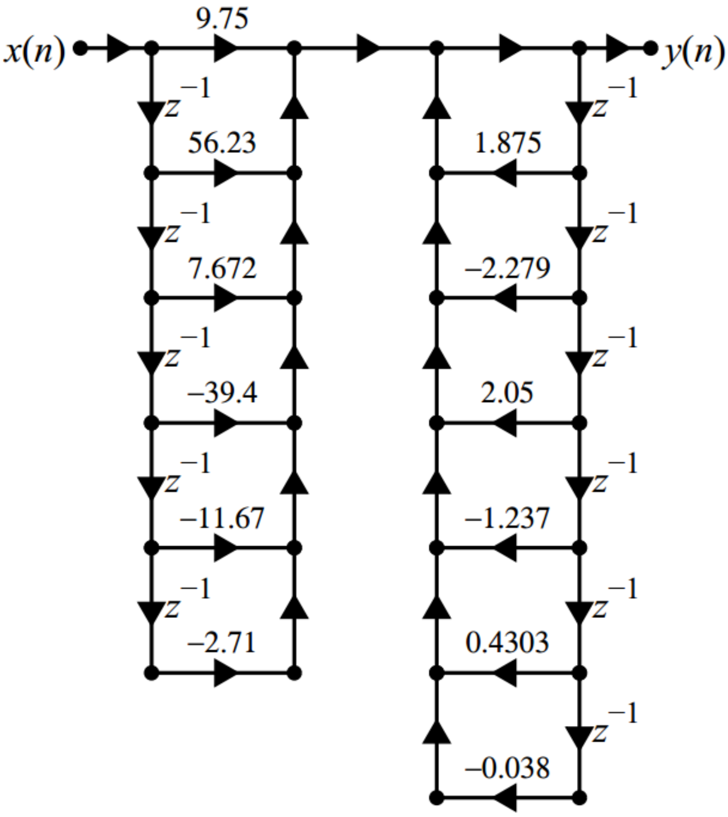
Determine and draw the following structures:

1. Normal direct form I
2. Normal direct form II
3. Cascade form containing transposed 2nd-order direct-form-II sections
4. Parallel form containing transposed 2nd-order direct-form-II sections
5. Lattice-ladder form

Solutions

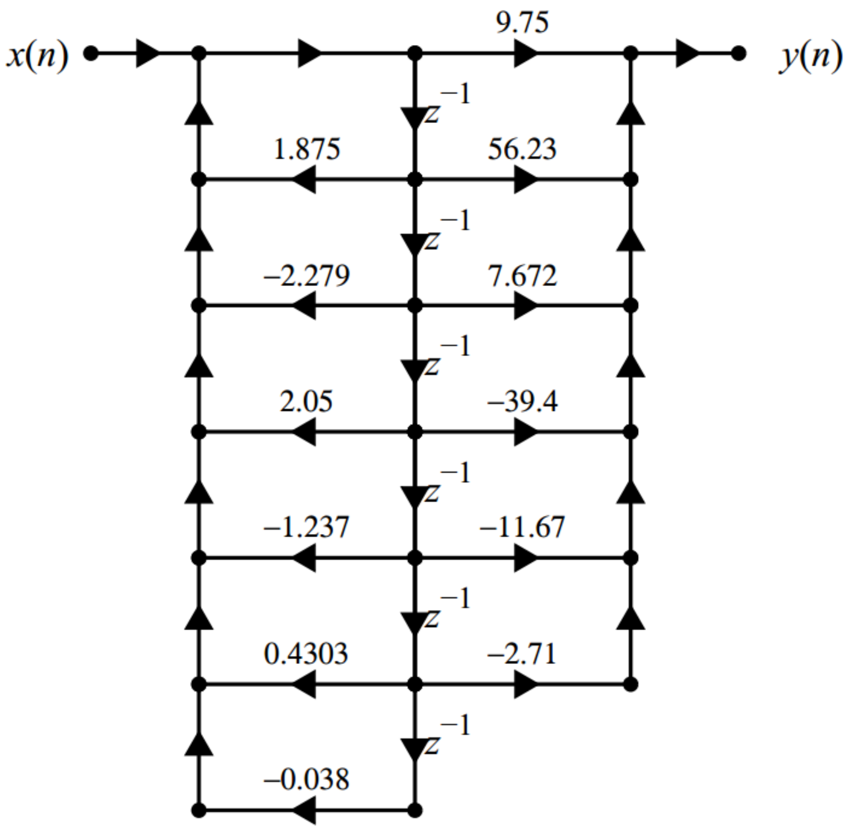
1. Normal direct form I: Matlab script:

```
% P6.8
%% P0608a.m
% (a) Normal Direct form-I
% Given H(z)
clear;
b1 = [-14.75,-12.9]; a1 = [1,-7/8,3/32];
b2 = [24.5,26.82,0]; a2 = [1,-1,1/2];
b3 = [1,2,1]; a3 = [1,0,0.81];
[b4,a4] = cas2dir(1,[b2;b3],[a2;a3]);
[C,B,A] = dir2par(b4,a4);
B = [B;b1]; A = [A;a1];
[b,a] = par2dir(C,B,A)

b =
    9.7500   56.2325    7.6719  -39.3959  -11.6666   -2.7101
0
a =
    1.0000   -1.8750    2.2788   -2.0500    1.2366   -0.4303
0.0380
```



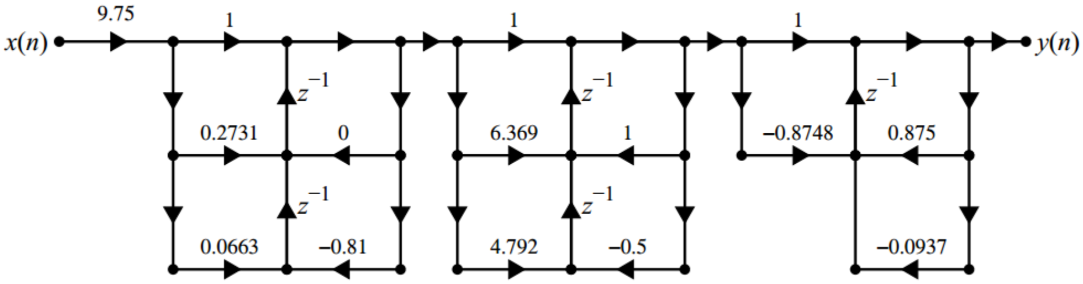
2. Normal direct form II: Matlab script:

```
%% P0608b.m
% (b) Normal Direct form-II
% Given H(z)
b1 = [-14.75,-12.9]; a1 = [1,-7/8,3/32];
b2 = [24.5,26.82,0]; a2 = [1,-1,1/2];
b3 = [1,2,1]; a3 = [1,0,0.81];
[b4,a4] = cas2dir(1,[b2;b3],[a2;a3]);
[C,B,A] = dir2par(b4,a4);
B = [B;b1]; A = [A;a1];
[b,a] = par2dir(C,B,A)

b =
    9.7500   56.2325    7.6719  -39.3959  -11.6666   -2.7101
0
a =
    1.0000   -1.8750    2.2788   -2.0500    1.2366   -0.4303
0.0380
```



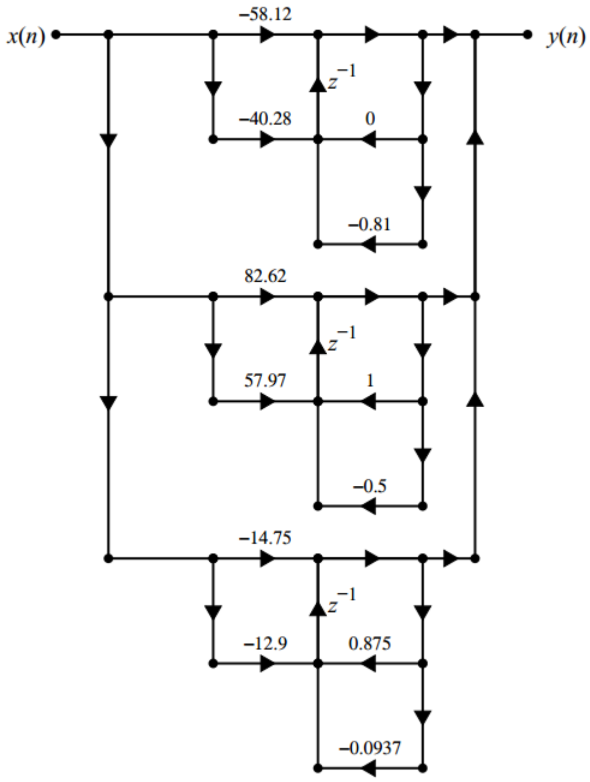
3. Cascade form containing transposed second-order direct form II sections: Matlab script:

```
%% P0608c.m
% (c) Transposed Cascade form
% Given H(z)
b1 = [-14.75,-12.9]; a1 = [1,-7/8,3/32];
b2 = [24.5,26.82,0]; a2 = [1,-1,1/2];
b3 = [1,2,1]; a3 = [1,0,0.81];
[b4,a4] = cas2dir(1,[b2;b3],[a2;a3]);
[C,B,A] = dir2par(b4,a4);
B = [B;b1]; A = [A;a1];
[b,a] = par2dir(C,B,A);
[b0,B,A] = dir2cas(b,a)

b0 =
    9.7500
B =
    1.0000    0.2731    0.0663
    1.0000    6.3691    4.7918
    1.0000   -0.8748         0
A =
    1.0000   -0.0000    0.8100
    1.0000   -1.0000    0.5000
    1.0000   -0.8750    0.0938
```



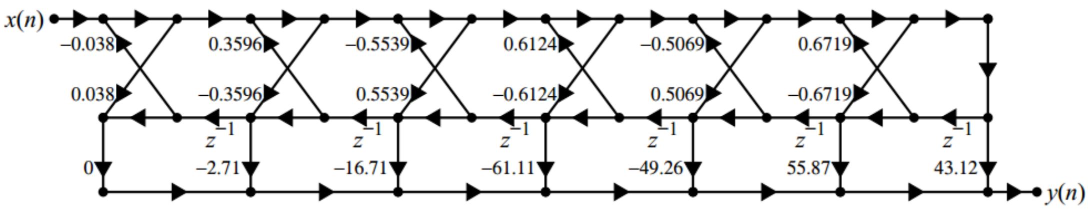
4. Parallel form containing transposed second-order direct form II sections: Matlab script:

```
%% P0608d.m
% (d) Transposed Parallel form
% Given H(z)
b1 = [-14.75,-12.9]; a1 = [1,-7/8,3/32];
b2 = [24.5,26.82,0]; a2 = [1,-1,1/2];
b3 = [1,2,1]; a3 = [1,0,0.81];
[b4,a4] = cas2dir(1,[b2;b3],[a2;a3]);
[C,B,A] = dir2par(b4,a4);
B = [B;b1]; A = [A;a1];
[b,a] = par2dir(C,B,A);
[C,B,A] = dir2par(b,a)

C =
     0
B =
  -58.1234  -40.2767
   82.6234   57.9733
  -14.7500  -12.9000
A =

    1.0000   -0.0000    0.8100
    1.0000   -1.0000    0.5000
    1.0000   -0.8750    0.0938
```



5. Lattice-ladder form: Matlab script:

```
%% P0608e.m
% Lattice-Ladder form
% Given H(z)
b1 = [-14.75,-12.9]; a1 = [1,-7/8,3/32];
b2 = [24.5,26.82,0]; a2 = [1,-1,1/2];
b3 = [1,2,1]; a3 = [1,0,0.81];
[b4,a4] = cas2dir(1,[b2;b3],[a2;a3]);
[C,B,A] = dir2par(b4,a4);
B = [B;b1]; A = [A;a1];
[b,a] = par2dir(C,B,A);
[K,C] = dir2ladr(b,a)

K =
   -0.6719    0.5069   -0.6124    0.5539   -0.3596    0.0380
C =
   43.1171   55.8710  -49.2642  -61.1147  -16.7111   -2.7101
0
```



P6.9

Figure P6.2 describes a causal linear time-invariant system. Determine and draw the following structures:

1. Direct form I
2. Direct form II
3. Cascade form containing second-order direct-form-II sections
4. Parallel form containing second-order direct-form-II sections

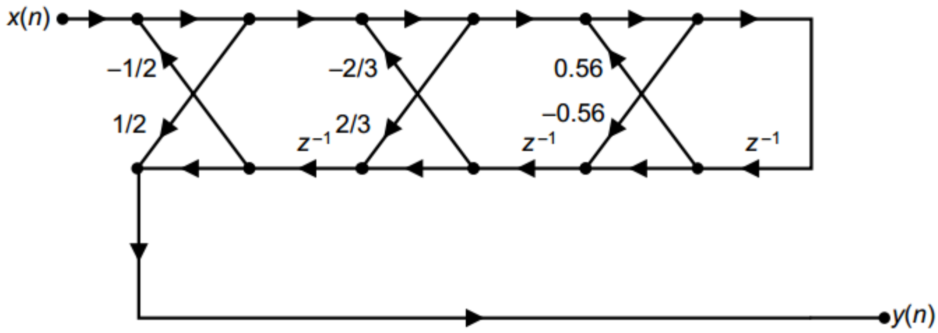
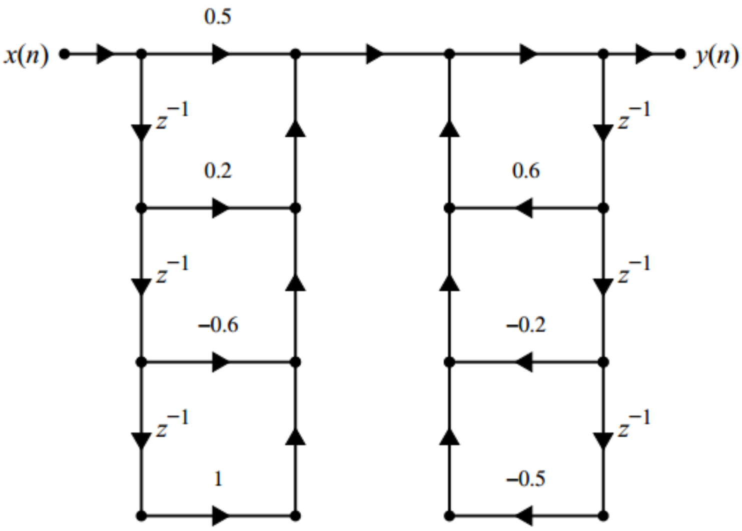


FIGURE P6.2 *Structure for Problem 6.9*

Solutions

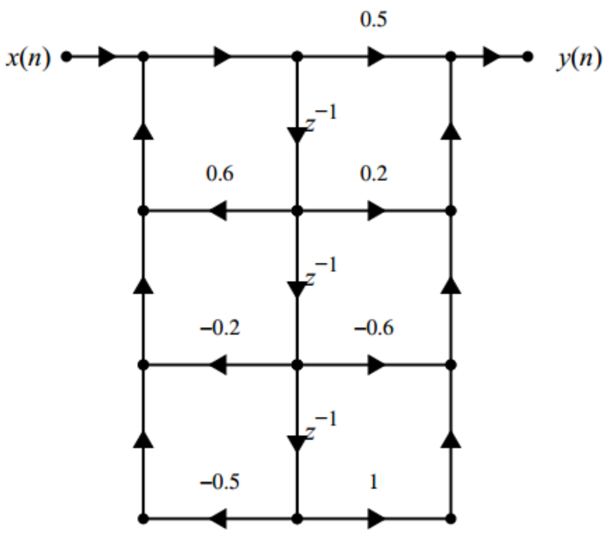
1. Direct form I: Given Lattice-Ladder

```
% P6.9
%% P0609a.m Direct form I: Given Lattice-Ladder
K = [-0.56,2/3,1/2]; C = [0,0,0,1]; [b,a] = ladr2dir(K,C)

b =
    0.5000    0.2000   -0.6000    1.0000
a =
    1.0000   -0.6000    0.2000    0.5000
```



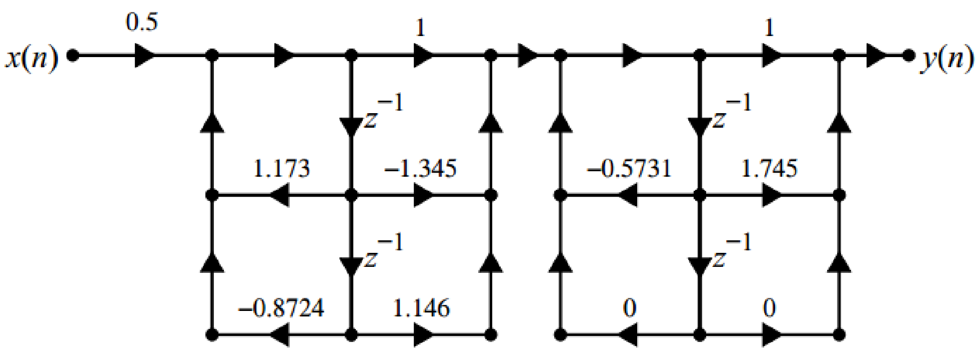
2. Direct form II:Given Lattice-Ladder

```
%% P0609b.m  Direct form II:Given Lattice-Ladder
K = [-0.56,2/3,1/2]; C = [0,0,0,1]; [b,a] = ladr2dir(K,C)
b =
    0.5000    0.2000   -0.6000    1.0000
a =
    1.0000   -0.6000    0.2000    0.5000
```



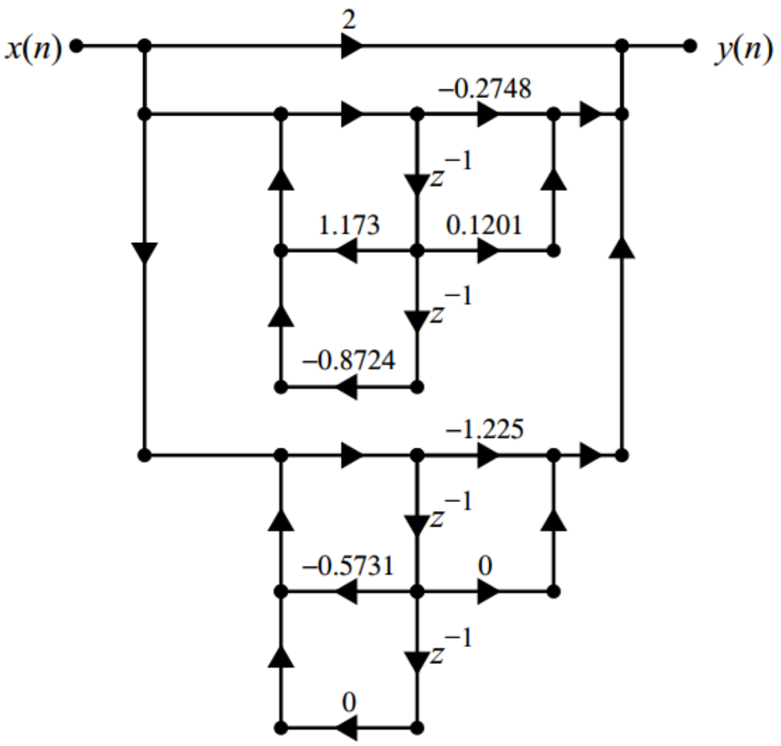
3. Cascade form containing second-order direct form II sections: Given Lattice-Ladder

```
%% P0609c.m Cascade form containing second-order direct
form II sections: Given Lattice-Ladder
K = [-0.56,2/3,1/2]; C = [0,0,0,1]; [b,a] =
ladr2dir(K,C);
[b0,B,A] = dir2cas(b,a)

b0 =
    0.5000
B =
    1.0000   -1.3448    1.1463
    1.0000    1.7448         0
A =
    1.0000   -1.1731    0.8724
    1.0000    0.5731         0
```



4. Parallel form containing second-order direct form II sections: Given Lattice-Ladder

```
%% P0609d.m Parallel form containing second-order direct
form II sections: Given Lattice-Ladder
K = [-0.56,2/3,1/2]; C = [0,0,0,1]; [b,a] =
ladr2dir(K,C);
[C,B,A] = dir2par(b,a)

C =
     2
B =
   -0.2748    0.1201
   -1.2252         0
A =
    1.0000   -1.1731    0.8724
    1.0000    0.5731         0
```



P6.10

A linear time-invariant system with system function

$$H(z) = \frac{0.05 - 0.01z^{-1} - 0.13z^{-2} + 0.13z^{-4} + 0.01z^{-5} - 0.05z^{-6}}{1 - 0.77z^{-1} + 1.59z^{-2} - 0.88z^{-3} + 1.2z^{-4} - 0.35z^{-5} + 0.31z^{-6}}$$

is to be implemented using a flow graph of the form shown in Figure P6.3.

1. Fill in all the coefficients in the diagram.

2. Is your solution unique? Explain

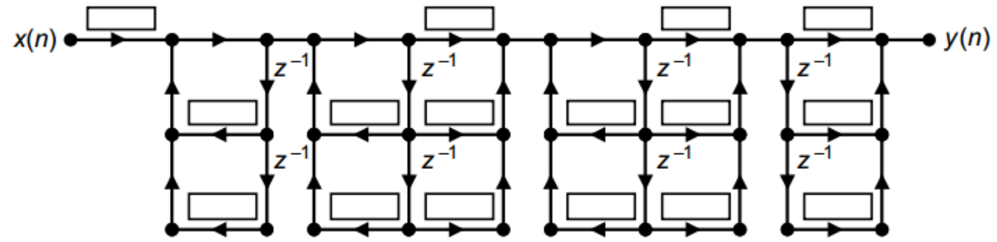


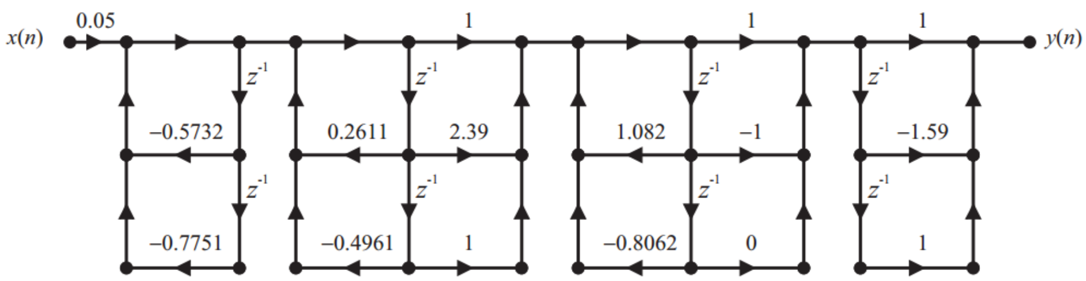
FIGURE P6.3 *Structure for Problem 6.10*

Solutions

1. The required form is the cascade form. Matlab script:

```
% P6.10
%The required form is the cascade form
b = [0.05,-0.01,-0.13,0.13,0.01,-0.05];
a = [1,-0.77,1.59,-0.88,1.2,-0.35,0.31];
[b0,B,A] = dir2cas(b,a)

b0 =
    0.0500
B =
    1.0000   -1.5900    1.0000
    1.0000    2.3900    1.0000
    1.0000   -1.0000         0
A =
    1.0000    0.5732    0.7751
    1.0000   -0.2611    0.4961
    1.0000   -1.0821    0.8062
```

Block diagram:



2. This solution is not unique since numerator and denominator biquads can be grouped differently.

P6.11

A linear time-invariant system with system function

$$H(z) = \frac{0.051 + 0.088z^{-1} + 0.06z^{-2} - 0.029z^{-3} - 0.069z^{-4} - 0.046z^{-5}}{1 - 1.34z^{-1} + 1.478z^{-2} - 0.789z^{-3} + 0.232z^{-4}}$$

is to be implemented using a flow graph of the form shown in Figure P6.4. Fill in all the coefficients in the diagram.

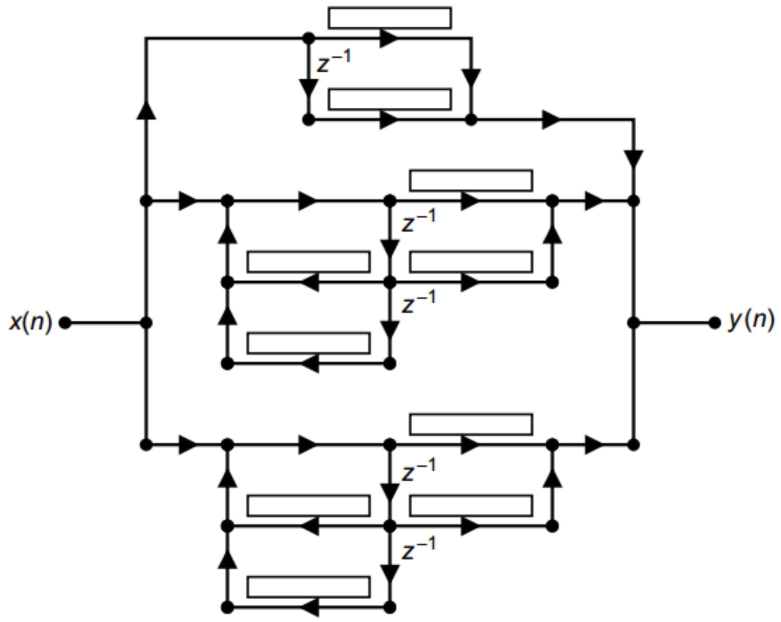


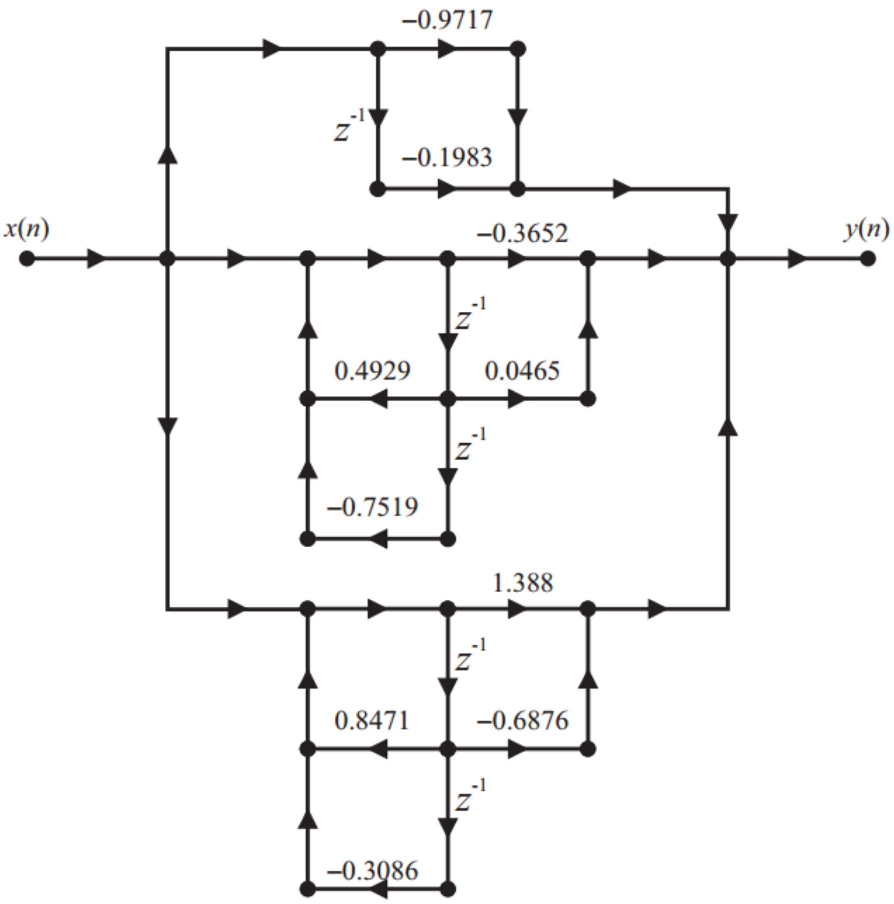
FIGURE P6.4 *Problem for Problem 6.11*

Solutions

The given flow graph is a parallel structure. Matlab script:

```
% P6.11
b = [0.051,0.088,0.06,-0.029,-0.069,-0.046];
a = [1,-1.34,1.478,-0.789,0.232];
[C,B,A] = dir2par(b,a)

C =
   -0.9717   -0.1983
B =
   -0.3654    0.0465
    1.3881   -0.6876
A =
    1.0000   -0.4929    0.7519
    1.0000   -0.8471    0.3086
```

Block diagram:



This solution is unique.

P6.12

Consider the linear time-invariant system given in Problem P6.10.

$$H(z) = \frac{0.05 - 0.01z^{-1} - 0.13z^{-2} + 0.13z^{-4} + 0.01z^{-5} - 0.05z^{-6}}{1 - 0.77z^{-1} + 1.59z^{-2} - 0.88z^{-3} + 1.2z^{-4} - 0.35z^{-5} + 0.31z^{-6}}$$

It is to be implemented using a flow graph of the form shown in Figure P6.5.

1. Fill in all the coefficients in the diagram.
2. Is your solution unique? Explain.

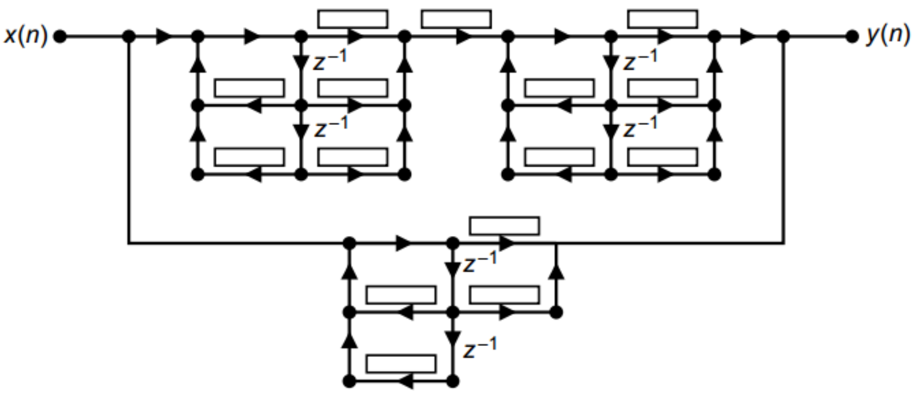


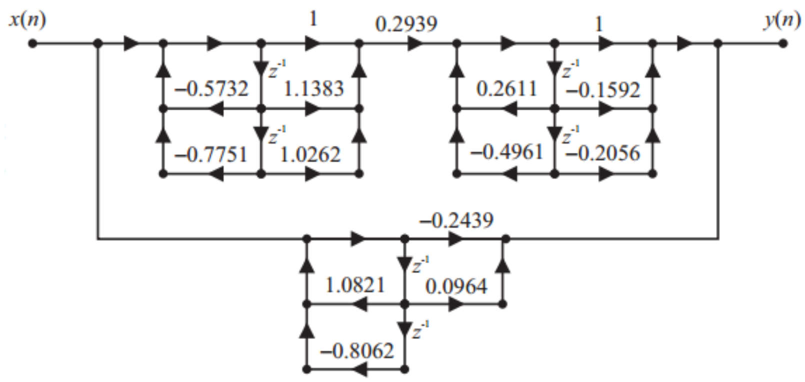
FIGURE P6.5 *Structure for Problem 6.12*

Solutions

1. The given signal flow graph is a parallel connection containing one second-order cascade branch. Matlab script:

```
% P6.12
% Given Direct form
b = [0.05,-0.01,-0.13,0,0.13,0.01,-0.05];
a = [1,-0.77,1.59,-0.88,1.2,-0.35,0.31];
% Convert to a parallel form
[C,B,A] = dir2par(b,a)

C =
  -0.1613
B =
  -0.2654   -0.0295
   0.7206   -0.1148
  -0.2439    0.0964
A =
   1.0000    0.5732    0.7751
   1.0000   -0.2611    0.4961
   1.0000   -1.0821    0.8062
% Convert the first two biquads into direct form
[b1,a1] = par2dir(C,B(1:2,:),A(1:2,:)); b1 = real(b1), a1
= real(a1)

b1 =
   0.2939    0.2878    0.1879   -0.1168   -0.0620
a1 =
   1.0000    0.3121    1.1215    0.0820    0.3845
% Convert the resulting direct into cascade form
[b0,B1,A1] = dir2cas(b1,a1)

b0 =
   0.2939
B1 =
   1.0000    1.1383    1.0262
   1.0000   -0.1592   -0.2056
A1 =
   1.0000    0.5732    0.7751
   1.0000   -0.2611    0.4961
```

Block diagram:



2. The solution is not unique since any two out of three parallel biquads can be used to construct a cascade branch.

P6.13

The filter structure shown in Figure P6.6 contains a parallel connection of cascade sections. Determine and draw the overall

1. direct form (normal) structure,
2. direct form (transposed) structure,
3. cascade form structure containing 2nd-order sections,
4. parallel form structure containing 2nd-order sections.

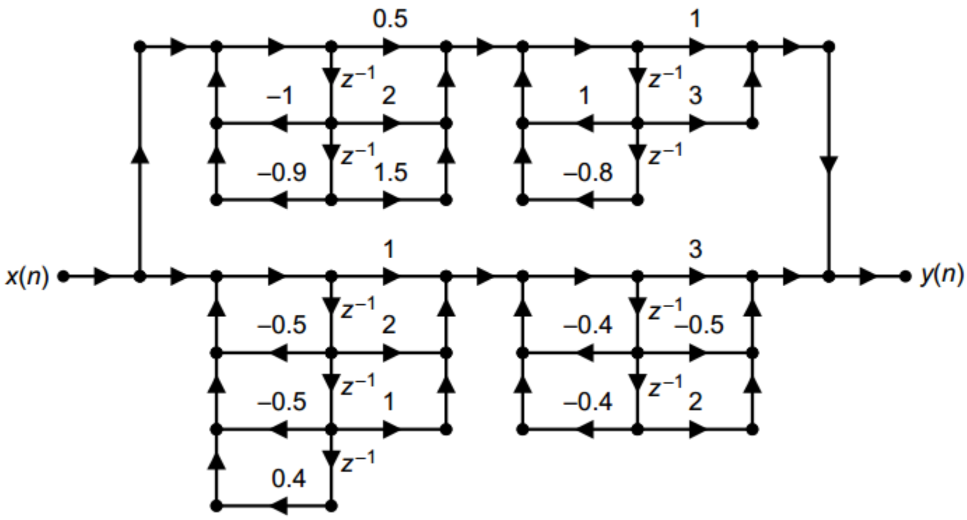


FIGURE P6.6 *Structure for Problem 6.13*

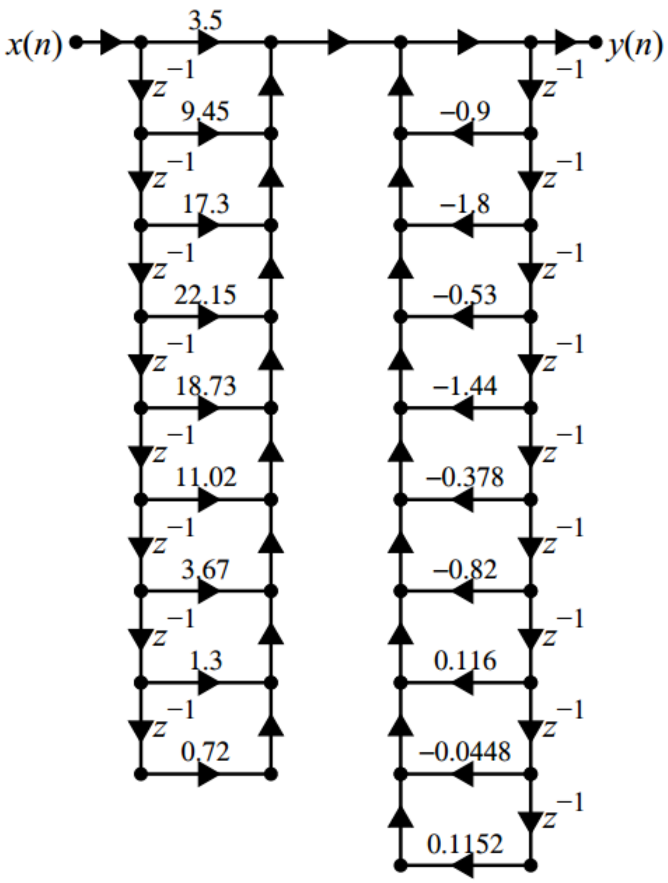
Solutions

1. Direct form (normal) structure: Matlab script:

```
% P6.13
%% P0613a.m
% (a) Normal Direct form-I
% Given Structure
```

```
% Upper parallel branch
[b1,a1] = cas2dir(1,[0.5,2,1.5;1,3,0],[1,1,0.9;1,-
1,0.8]);
b1 = removetrailzeros(b1); [C1,B1,A1] = dir2par(b1,a1);
% Lower parallel branch
[b0,B2,A2] = dir2cas([1,2,1],[1,0.5,0.5,-0.4]);
[b2,a2] = cas2dir(b0,[3,-0.5,2;B2],[1,0.4,0.4;A2]);
b2 = removetrailzeros(b2); a2 = removetrailzeros(a2);
[C2,B2,A2] = dir2par(b2,a2);
% Overall parallel
C = C1+C2; B = [B1;B2]; A = [A1;A2];
% Overall direct
[b,a] = par2dir(C,B,A); b = real(b), a = real(a)
b = Columns 1 through 9
3.5000 9.4500 17.3000 22.1500 18.7300 11.0200 3.6700
1.3000 0.7200
a = Columns 1 through 10
1.0000 0.9000 1.8000 0.5300 1.4400 0.3780 0.8200 -0.1160
0.0448 0.1152
```

Block diagram:

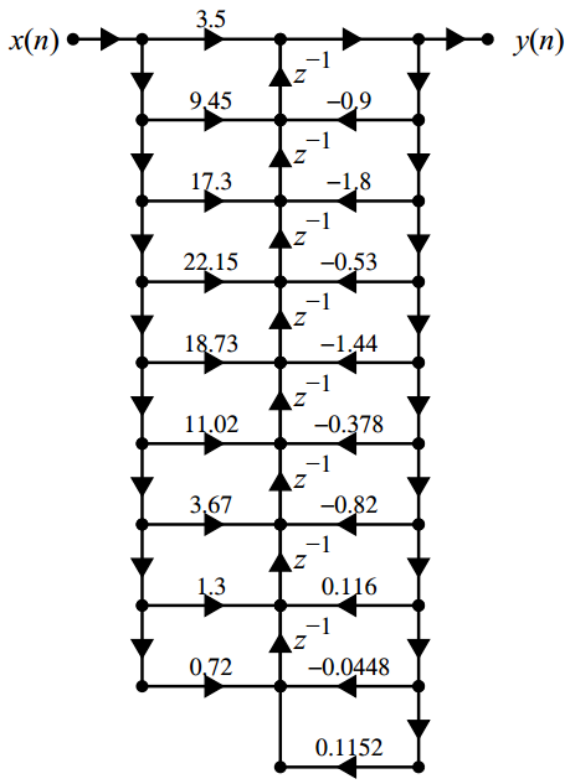


2. Direct form (transposed) structure: Matlab script:

```
%% P0613b.m
```

```
% (b) Transposed Direct form-II
% Given Structure
% Upper parallel branch
clear;
[b1,a1] = cas2dir(1,[0.5,2,1.5;1,3,0],[1,1,0.9;1,-
1,0.8]);
b1 = removetrailzeros(b1); [C1,B1,A1] = dir2par(b1,a1);
% Lower parallel branch
[b0,B2,A2] = dir2cas([1,2,1],[1,0.5,0.5,-0.4]);
[b2,a2] = cas2dir(b0,[3,-0.5,2;B2],[1,0.4,0.4;A2]);
b2 = removetrailzeros(b2); a2 = removetrailzeros(a2);
[C2,B2,A2] = dir2par(b2,a2);
% Overall parallel
C = C1+C2; B = [B1;B2]; A = [A1;A2];
% Overall direct
[b,a] = par2dir(C,B,A); b = real(b), a = real(a)

b = Columns 1 through 9
3.5000 9.4500 17.3000 22.1500 18.7300 11.0200 3.6700
1.3000 0.7200
a = Columns 1 through 10
1.0000 0.9000 1.8000 0.5300 1.4400 0.3780 0.8200 -0.1160
0.0448 0.1152
```

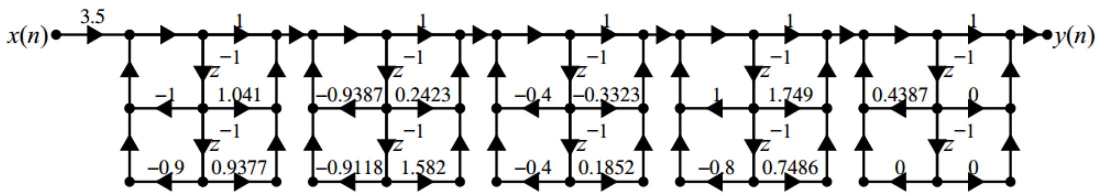
Block diagram:



3. Cascade form structure containing second-order sections: Matlab script:

```
%% P0613c.m
% (c) Normal Cascade form
% Given Structure
% Upper parallel branch
clear;
[b1,a1] = cas2dir(1,[0.5,2,1.5;1,3,0],[1,1,0.9;1,-
1,0.8]);
[C1,B1,A1] = dir2par(b1,a1);
% Lower parallel branch
[b0,B2,A2] = dir2cas([1,2,1],[1,0.5,0.5,-0.4]);
[b2,a2] = cas2dir(b0,[3,-0.5,2;B2],[1,0.4,0.4;A2]);
b2 = removetrailzeros(b2); a2 = removetrailzeros(a2);
[C2,B2,A2] = dir2par(b2,a2);
% Overall parallel
C = C1+C2; B = [B1;B2]; A = [A1;A2];
% Overall direct
[b,a] = par2dir(C,B,A); b = real(b); a = real(a);
[b0,Bc,Ac] = dir2cas(b,a)

b0 =
3.5000
Bc =
1.0000 1.0414 0.9377
1.0000 0.2423 1.5819
1.0000 -0.3323 0.1852
1.0000 1.7486 0.7486
1.0000 0.0000 0
Ac =
1.0000 1.0000 0.9000
1.0000 0.9387 0.9118
1.0000 0.4000 0.4000
1.0000 -1.0000 0.8000
1.0000 -0.4387 0.0000
```

Block diagram:

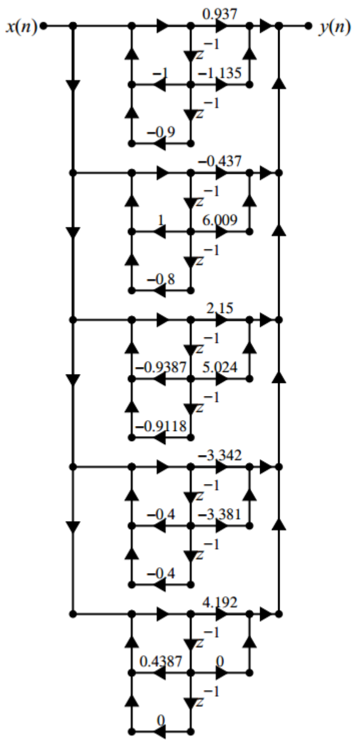


4. Parallel form structure containing second-order sections:

```
%% P0613d.m
% (d) Normal Parallel form
% Given Structure
```

```
% Upper parallel branch
clear;
[b1,a1] = cas2dir(1,[0.5,2,1.5;1,3,0],[1,1,0.9;1,-
1,0.8]);
[C1,B1,A1] = dir2par(b1,a1);
% Lower parallel branch
[b0,B2,A2] = dir2cas([1,2,1],[1,0.5,0.5,-0.4]);
[b2,a2] = cas2dir(b0,[3,-0.5,2;B2],[1,0.4,0.4;A2]);
b2 = removetrailzeros(b2); a2 = removetrailzeros(a2);
[C2,B2,A2] = dir2par(b2,a2);
% Overall parallel
C = C1+C2, B = [B1;B2], A = [A1;A2]

C =
     []
B =
    0.9370   -1.1349
   -0.4370    6.0088
    2.1502    5.0236
   -3.3424   -3.3813
    4.1923         0
A = 1.0000    1.0000    0.9000
    1.0000   -1.0000    0.8000
    1.0000    0.9387    0.9118
    1.0000    0.4000    0.4000
    1.0000   -0.4387         0
```

Block diagram:



P6.14

In filter structure shown in Figure P6.7, systems $H_1(z)$ and $H_2(z)$ are subcomponents of a larger system *H*(*z*). The system function $H_1(z)$ is given in the parallel form

$$H_1(z) = 2 + \frac{0.2 - 0.3z^{-1}}{1 + 0.9z^{-1} + 0.9z^{-2}} + \frac{0.4 + 0.5z^{-1}}{1 - 0.8z^{-1} + 0.8z^{-2}}$$

and the system function $H_2(z)$ is given in the cascade form

$$H_2(z) = \left(\frac{2 + z^{-1} - z^{-2}}{1 + 1.7z^{-1} + 0.72z^{-2}}\right)\left(\frac{3 + 4z^{-1} + 5z^{-2}}{1 - 1.5z^{-1} + 0.56z^{-2}}\right)$$

1. Express *H*(*z*) as a rational function.
2. Draw the block diagram of *H*(*z*) as a cascade-form structure.
3. Draw the block diagram of *H*(*z*) as a parallel-form structure.

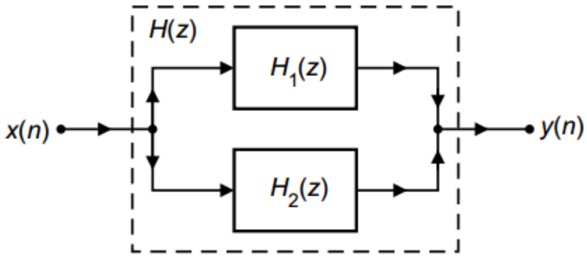


FIGURE P6.7 *Structure for Problem 6.14*

Solutions

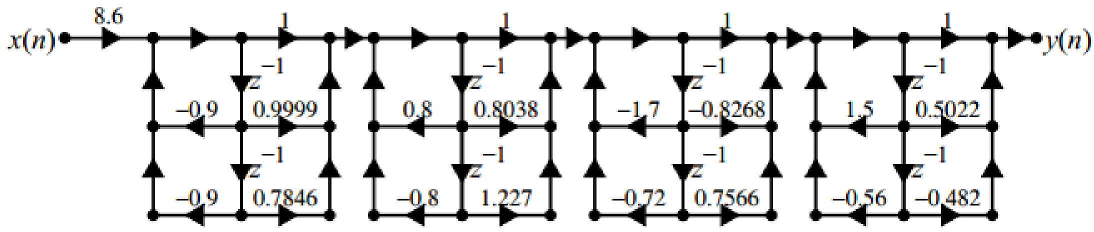
1. *H*(*z*) as a rational function: Matlab script:

```
%% P0614a.m
clear;clc;
% Given Structure
% H1(z) in parallel form: Leave as is
C1 = 2; B1 = [0.2,-0.3;0.4,0.5]; A1 = [1,0.9,0.9;1,-
0.8,0.8];
% H2(z) in cascade form: Convert to parallel form
b0 = 1; B2 = [2,1,-1;3,4,5]; A2 = [1,1.7,0.72;1,-
1.5,0.56];
[b2,a2] = cas2dir(b0,B2,A2); [C2,B2,A2] = dir2par(b2,a2);
% Combine two parallel forms
C = C1+C2; B = [B1;B2]; A = [A1;A2];
% (a) Rational function H(z)
[b,a] = par2dir(C,B,A); b = real(b), a = real(a)
```

```
b =
  Columns 1 through 8
    8.6000   12.7200   17.9680   12.6292    8.6276    8.2575
2.4425    0.6204
  Column 9
   -3.0194
a =
  Columns 1 through 8
    1.0000    0.3000   -0.2700   -0.0590   -0.1342    0.0589
-0.5193   -0.0922
  Column 9
    0.2903
```

2. Cascade form structure:

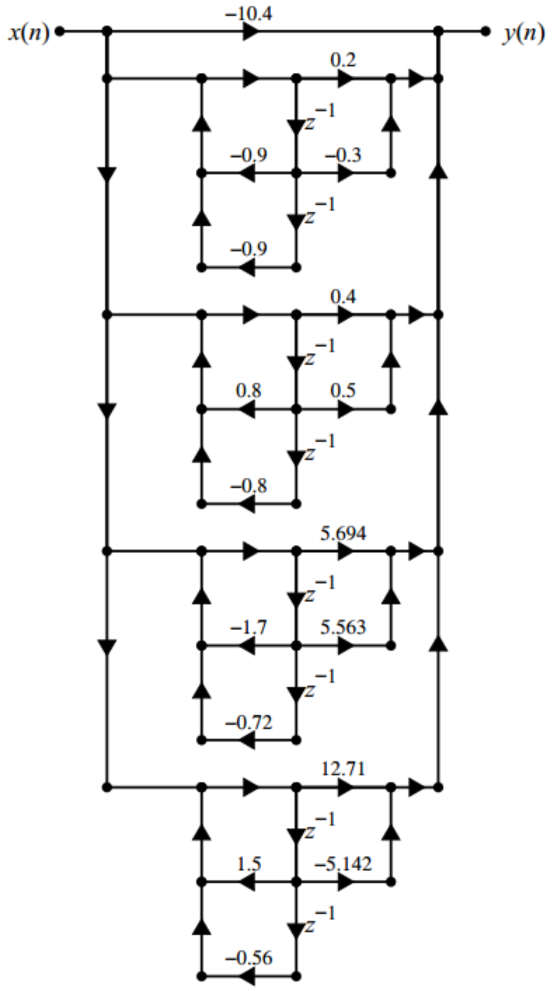
```
%% P0614b.m
% Cascade form structure:
[b0,Bc,Ac] = dir2cas(b,a)

b0 =
    8.6000
Bc =
    1.0000    0.9999    0.7846
    1.0000    0.8038    1.2272
    1.0000   -0.8268    0.7566
    1.0000    0.5022   -0.4820
Ac =
    1.0000    0.9000    0.9000
    1.0000   -0.8000    0.8000
    1.0000    1.7000    0.7200
    1.0000   -1.5000    0.5600
```



3. Parallel form structure:

```
%% P0614c.m
% Parallel form structure:
[C,B,A] = dir2par(b,a)

C =
  -10.4008
B =
    0.2000   -0.3000
```

```
    0.4000    0.5000
    5.6943    5.5629
   12.7065   -5.1424
A =
    1.0000    0.9000    0.9000
    1.0000   -0.8000    0.8000
    1.0000    1.7000    0.7200
    1.0000   -1.5000    0.5600
```



P6.15

The digital filter structure shown in Figure P6.8 is a cascade of 2 parallel sections and corresponds to a 10th-order IIR digital filter system function

$$H(z) = \frac{1 - 2.2z^{-2} + 1.6368z^{-4} - 0.48928z^{-6} + 5395456 \times 10^{-8}z^{-8} - 147456 \times 10^{-8}z^{-10}}{1 - 1.65z^{-2} + 0.8778z^{-4} - 0.17281z^{-6} + 1057221 \times 10^{-8}z^{-8} - 893025 \times 10^{-10}z^{-10}}$$

1. Due to an error in labeling, two of the multiplier coefficients (rounded to 4 decimals) in this structure have incorrect values. Locate these 2 multipliers and determine their correct values.
2. Determine and draw an overall cascade structure containing 2nd-order section and which contains the least number of multipliers.

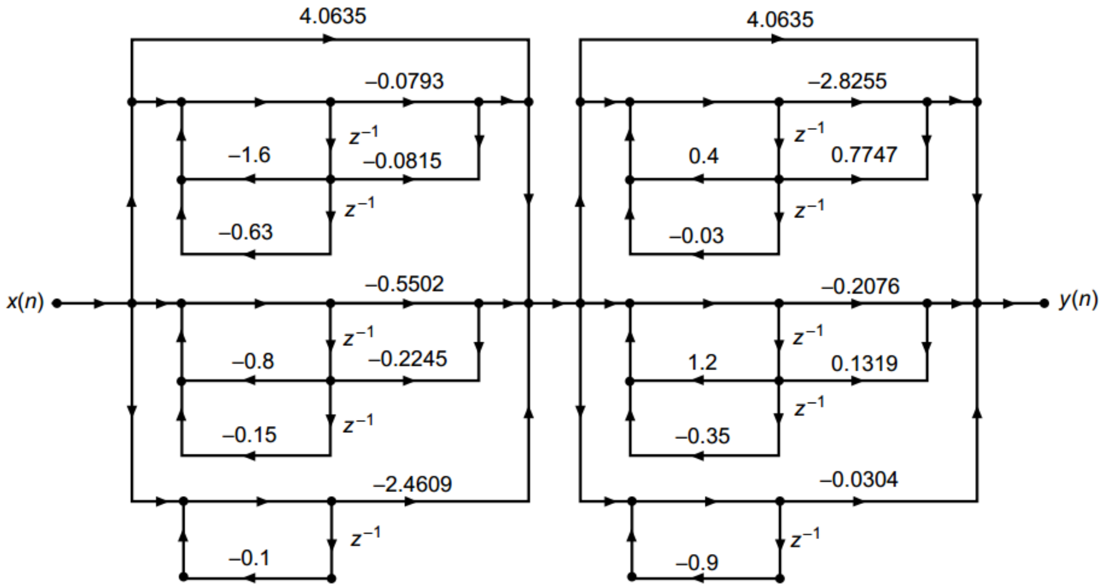


FIGURE P6.8 *Structure for Problem 6.15*

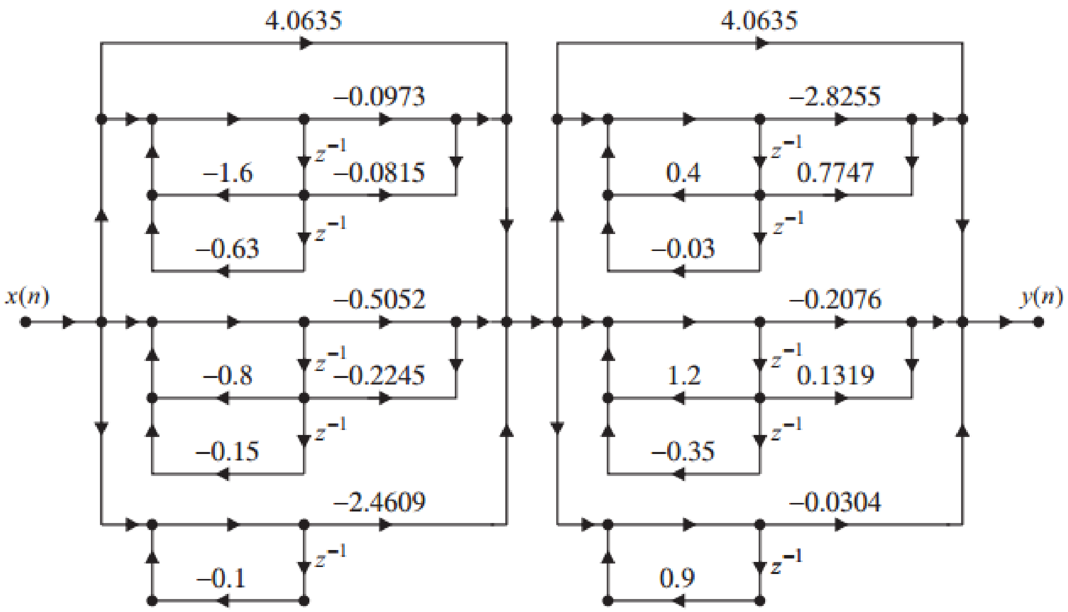
Solutions

1. Due to a mistake in labeling, two of the multiplier coefficients in this structure have ***incorrect values*** (rounded to 4 decimals). To locate these two multipliers and determine their correct values we will investigate their pole-zero structure and combine the appropriate pairs. Matlab Script:

```
%% P0615a.m
clc; format short;
% Given Rational Function
b = [1,0,-2.2,0,1.6368,0,-0.48928,0,5395456e-8,0,-
147456e-8];
a = [1,0,-1.65,0,0.8778,0,-0.17281,0,1057221e-8,0,-
893025e-10];
% Poles/Zeros of the System function
zs = roots(b); zs = sort(zs); ps = roots(a); ps =
sort(ps);
% Poles and zeros chosen from each half to create two
parallel sections
% Parallel Section-1
b3 = poly(zs([1:5])); a3 = poly(ps([1:5])); [C3,B3,A3] =
dir2par(b3,a3)

C3 =
    4.0635
B3 =
   -0.0973   -0.0815
```

```
   -0.5052   -0.2245
   -2.4609         0
A3 =
    1.0000    1.6000    0.6300
    1.0000    0.8000    0.1500
    1.0000    0.1000         0
% Parallel Section-2
b4 = poly(zs([6:10])); a4 = poly(ps([6:10])); [C4,B4,A4]
= dir2par(b4,a4)

C4 =
    4.0635
B4 =
   -2.8255    0.7747
   -0.2076    0.1319
   -0.0304         0
A4 =
    1.0000   -0.4000    0.0300
    1.0000   -1.2000    0.3500
    1.0000   -0.9000         0
```

Hence from inspection, the two incorrect values are -0.5502 in the first parallel section and -0.9 in the second parallel section. The correct block diagram is:

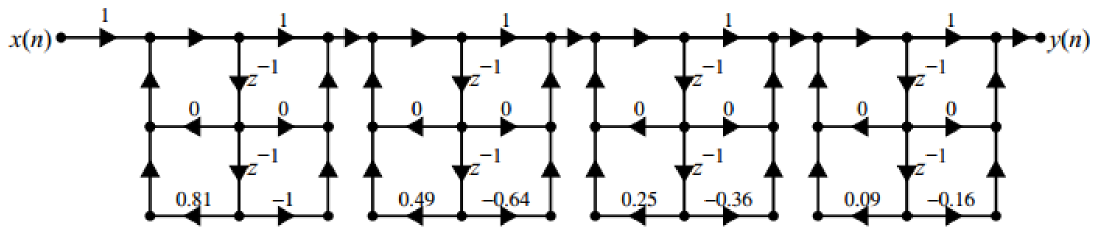


2. An overall cascade structure containing second-order section and which contains the least number of multipliers: This can be obtained by combining pole or zero pairs with the same magnitude but differing signs. Matlab script:

```
%% P0615b.m
% (b) Overall Cascade structure containing the least
number of multipliers
```

```
% Combine the poles and zeros with the same magnitude but
+- signs.
b0 = 1;
B = [poly(zs([1,end]));poly(zs([2,end-
1]));poly(zs([3,end-2]));...poly(zs([4,end-
3]));poly(zs([5,end-4]))]
A = [poly(ps([1,end]));poly(ps([2,end-
1]));poly(ps([3,end-2]));...poly(ps([4,end-3]))]

B =
    1.0000         0   -1.0000
    1.0000    0.0000   -0.6400
    1.0000   -0.0000   -0.3600
    1.0000   -0.0000   -0.1600
    1.0000   -0.0000   -0.0400
A =
    1.0000   -0.0000   -0.8100
    1.0000    0.0000   -0.4900
    1.0000   -0.0000   -0.2500
    1.0000    0.0000   -0.0900
```

The following block diagram has only 7 multipliers (not counting multiplication by 0, 1, or -1)



P6.16

As described in this chapter, a linear-phase FIR filter is obtained by requiring certain symmetry conditions on its impulse responses.

1. In the case of symmetrical impulse response, we have $h(n) = h(M - 1 - n)$, $0 \leq n \leq M - 1$. Show that the resulting phase response is linear in ω and is given by

$$\angle H\left(e^{j\omega}\right) = -\left(\frac{M-1}{2}\right)\omega, \quad -\pi < \omega \leq \pi$$

2. Draw the linear-phase structures for this form when $M = 5$ and $M = 6$.

3. In the case of antisymmetrical impulse response, we have $h(n) = -h(M - 1 - n)$, $0 \leq n \leq M - 1$. Show that the resulting phase response is given by

$$\angle H\left(e^{j\omega}\right) = \pm\frac{\pi}{2} - \left(\frac{M-1}{2}\right)\omega, \quad -\pi < \omega \leq \pi$$

4. Draw the linear-phase structures for this form when $M = 5$ and $M = 6$.

Solutions

1. In the case of symmetrical impulse response, we have $h(n) = h(M - 1 - n)$, $0 \le n \le M - 1$. If M is an odd number then

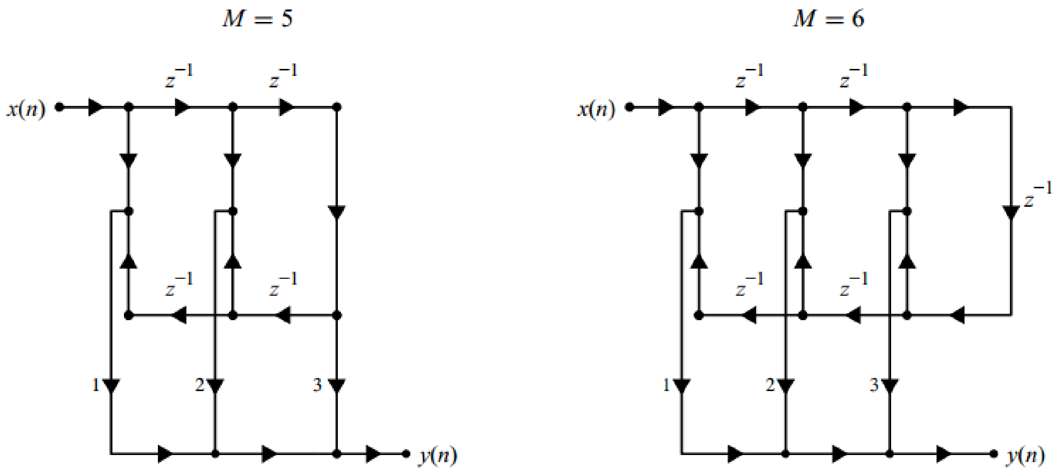
$$\begin{aligned} H(e^{j\omega}) &= \sum_{0}^{M-1} h(n)\, e^{-j\omega n} = \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega n} + h\left(\frac{M-1}{2}\right) e^{-j\omega\left(\frac{M-1}{2}\right)} + \sum_{\frac{M+1}{2}}^{M-1} h(n)\, e^{-j\omega n} \\ &= \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega n} + h\left(\frac{M-1}{2}\right) e^{-j\omega\left(\frac{M-1}{2}\right)} + \sum_{\frac{M+1}{2}}^{M-1} h(M-1-n)\, e^{-j\omega n} \\ &= \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega n} + h\left(\frac{M-1}{2}\right) e^{-j\omega\left(\frac{M-1}{2}\right)} + \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega(M-1-n)} \\ &= e^{-j\omega\left(\frac{M-1}{2}\right)} \left[\sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega\left(n-\frac{M-1}{2}\right)} + h\left(\frac{M-1}{2}\right) + \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{j\omega\left(n-\frac{M-1}{2}\right)} \right] \\ &= e^{-j\omega\left(\frac{M-1}{2}\right)} \left[h\left(\frac{M-1}{2}\right) + \sum_{0}^{\frac{M-3}{2}} 2h(n) \cos\left[\left(n - \frac{M-1}{2}\right)\right] \right] \end{aligned} \tag{6.5}$$

Since the term in the bracket on the right in (6.5) is real-valued, the resulting phase response is given by the first term. It is linear in ω and is given by

$$\angle H\left(e^{j\omega}\right) = -\left(\frac{M-1}{2}\right)\omega, \quad -\pi < \omega \le \pi \tag{6.6}$$

Similarly, if M is an even number then the term $h\left(\frac{M-1}{2}\right) = 0$ in (6.5) and instead of the term $(M - 3)/2$, we get the term $(M/2 - 1)$ in (6.5). Hence the phase response is still linear and is given by (6.6).

2. Linear-phase structures for the symmetric form when $M = 5$ and $M = 6$:



3. In the case of antisymmetrical impulse response, we have $h(n) = -h(M - 1 - n)$, $0 \le n \le M - 1$.

If M is an odd number then $h\left(\frac{M-1}{2}\right) = 0$. Hence

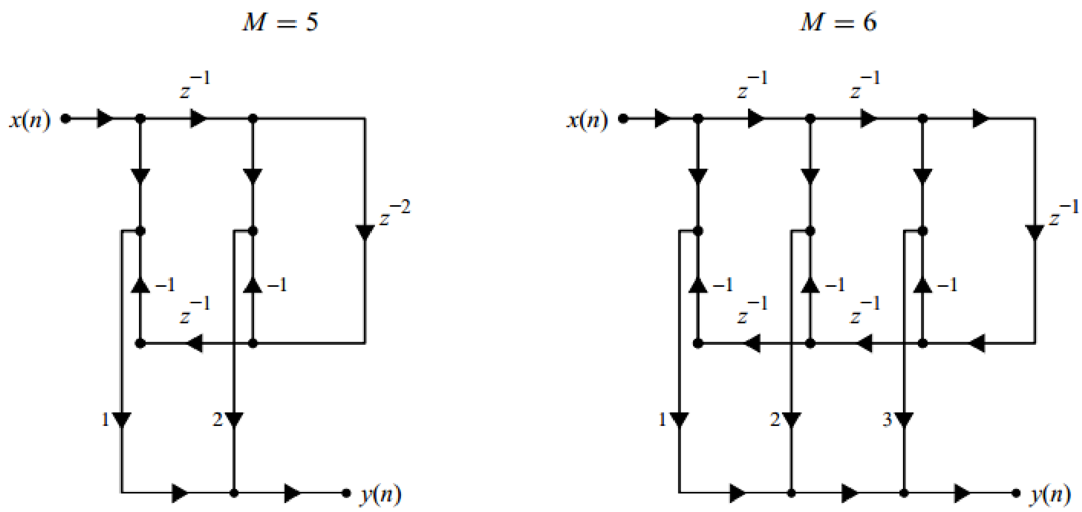
$$
\begin{aligned}
H(e^{j\omega}) &= \sum_{0}^{M-1} h(n)\, e^{-j\omega n} = \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega n} + \sum_{\frac{M+1}{2}}^{M-1} h(n)\, e^{-j\omega n} \\
&= \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega n} - \sum_{\frac{M+1}{2}}^{M-1} h(M-1-n)\, e^{-j\omega n} = \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega n} - \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega(M-1-n)} \\
&= e^{-j\omega\left(\frac{M-1}{2}\right)} \left[\sum_{0}^{\frac{M-3}{2}} h(n)\, e^{-j\omega\left(n-\frac{M-1}{2}\right)} - \sum_{0}^{\frac{M-3}{2}} h(n)\, e^{j\omega\left(n-\frac{M-1}{2}\right)} \right] \\
&= e^{-j\omega\left(\frac{M-1}{2}\right)} \left[-j \sum_{0}^{\frac{M-3}{2}} 2h(n) \sin\left\{ \left(n - \frac{M-1}{2} \right) \right\} \right] \\
&= e^{j\left[\pm\frac{\pi}{2} - \left(\frac{M-1}{2}\right)\omega\right]} \sum_{0}^{\frac{M-3}{2}} 2h(n) \sin\left\{ \left(n - \frac{M-1}{2} \right) \right\} \qquad (6.7)
\end{aligned}
$$

Again the term in the bracket on the right in (6.7) is real-valued and hence the resulting phase response is given by the first term. It is a linear equation in ω and is given by

$$
\angle H\left(e^{j\omega}\right) = \pm\frac{\pi}{2} - \left(\frac{M-1}{2}\right)\omega, \quad -\pi < \omega \leq \pi \qquad (6.8)
$$

Similarly, if M is an even number then instead of the term $(M - 3)/2$, we get the term $(M/2 - 1)$ in (6.7). Hence the phase response is still linear and is given by (6.8).

4. Linear-phase structures for the antisymmetric form when $M = 5$ and $M = 6$:



P6.17

An FIR filter is described by the difference equation

$$
y(n) = \sum_{k=0}^{6} e^{-0.9|k-3|} x(n-k)
$$

Determine and draw the block diagrams of the following structures.

1. Direct form

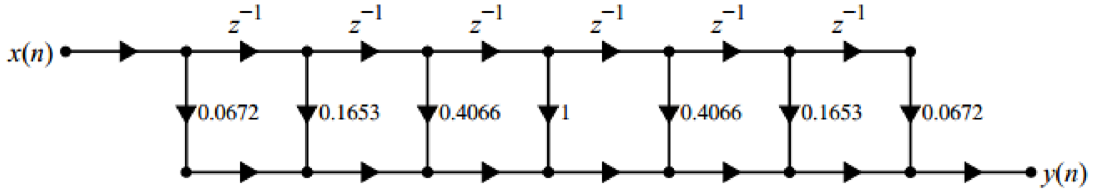
2. Linear-phase form
3. Cascade form
4. Frequency sampling form

Solutions

1. Direct form: Matlab script:

```
% P6.17
%% P0617a.m
% (a) Direct form
% Given FIR filter coefficients
k = [0:6]; b = exp(-0.9*abs(k-3))

b =
    0.0672    0.1653    0.4066    1.0000    0.4066    0.1653
0.0672
```

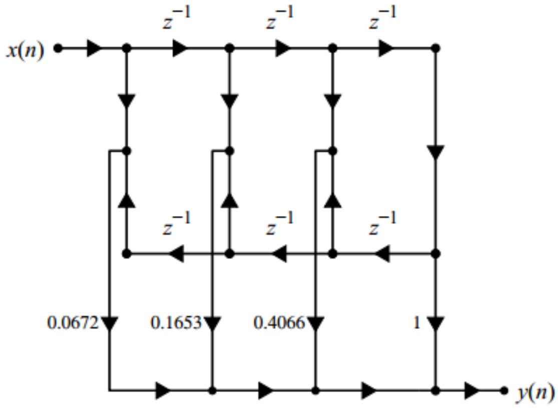
The block diagram is:



2. Linear-phase form: Matlab script:

```
%% P0617b.m
% (b) Linear-phase form
% Given FIR filter coefficients
k = [0:6]; b = exp(-0.9*abs(k-3))

b =
    0.0672    0.1653    0.4066    1.0000    0.4066    0.1653
0.0672
```

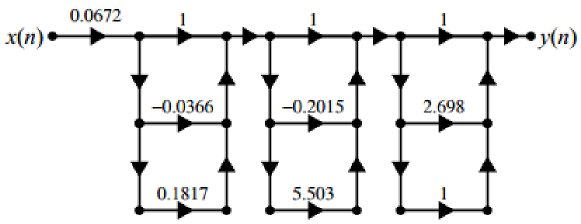
The block diagram is:



3. Cascade form: Matlab script:

```
%% P0617c.m
% (c) Cascade Form
% Given FIR filter coefficients
k = [0:6]; b = exp(-0.9*abs(k-3)); a = 1;
[b0,B,A] = dir2cas(b,a)

b0 =
    0.0672
B =
    1.0000   -0.0366    0.1817
    1.0000   -0.2015    5.5030
    1.0000    2.6978    1.0000
A =
     1     0     0
     1     0     0
     1     0     0
```

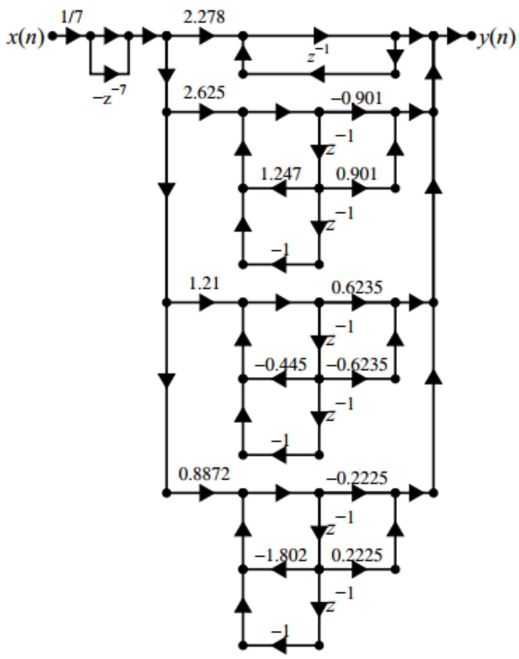
The block diagram is:



4. Frequency sampling form: Matlab script:

```
%% P0617d.m
% (d) Frequency-sampling form
% Given FIR filter coefficients
k = [0:6]; b = exp(-0.9*abs(k-3));
[C,B,A] = dir2fs(b)
C =
    2.6246
    1.2100
    0.8872
    2.2781
B =
   -0.9010    0.9010
    0.6235   -0.6235
   -0.2225    0.2225
A =
    1.0000   -1.2470    1.0000
    1.0000    0.4450    1.0000
    1.0000    1.8019    1.0000
    1.0000   -1.0000         0
```

The block diagram is:



P6.18

A linear time-invariant system is given by the system function

$$H(z) = 2 + 3z^{-1} + 5z^{-2} - 3z^{-3} + 4z^{-5} + 8z^{-7} - 7z^{-8} + 4z^{-9}$$

Determine and draw the block diagrams of the following structures.

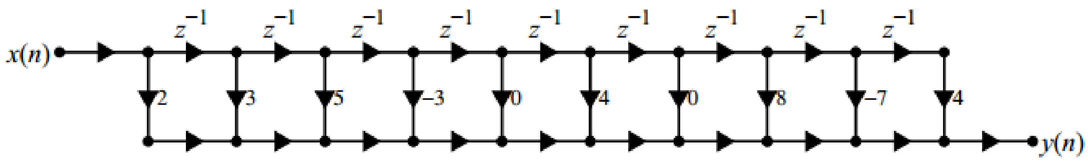
1. Direct form
2. Cascade form
3. Lattice form
4. Frequency sampling form

Solutions

1. Direct form: Matlab script:

```
%% P0618a.m
% (a) Direct form
% Given FIR filter coefficients
b = [2,3,5,-3,0,4,0,8,-7,4]; a = 1;
```

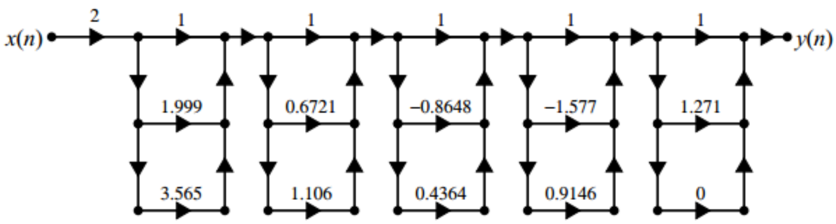
The block diagram is:



2. Cascade form: Matlab script:

```
%% P0618b.m
% (b) Cascade form
% Given FIR filter coefficients
b = [2,3,5,-3,0,4,0,8,-7,4]; a = 1;
[b0,B,A] = dir2cas(b,a)

b0 =
     2
B =
    1.0000    1.9987    3.5650
    1.0000    0.6721    1.1062
    1.0000   -0.8648    0.4364
    1.0000   -1.5766    0.9146
    1.0000    1.2706         0
A =
     1     0     0
     1     0     0
     1     0     0
     1     0     0
     1     0     0
```

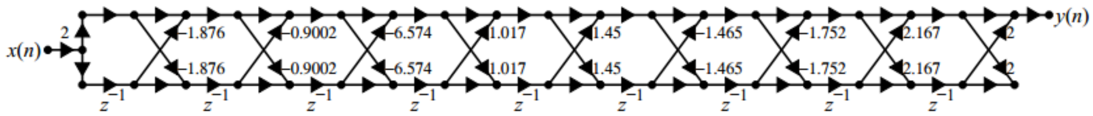
The block diagram is:



3. Lattice form: Matlab script:

```
%% P0618c.m
% (c) Lattice form
% Given FIR filter coefficients
b = [2,3,5,-3,0,4,0,8,-7,4]; a = 1;
[K] = dir2latc(b)

K =
  Columns 1 through 8
    2.0000   -1.8759   -0.9002   -6.5739    1.0170    1.4501
-1.4653   -1.7519
  Columns 9 through 10
    2.1667    2.0000
```

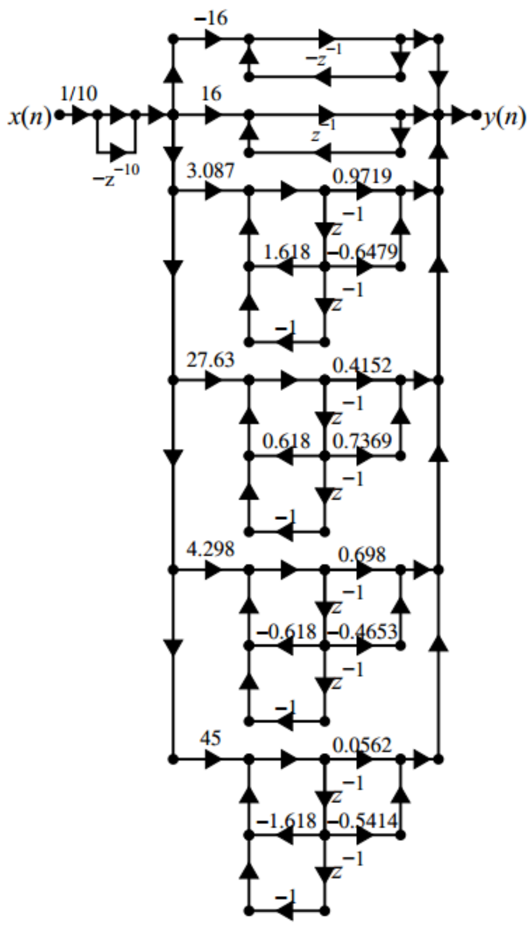
The block diagram is:



4. Frequency sampling form: Matlab script:

```
%% P0618d.m
% (d) Frequency-sampling form
% Given FIR filter coefficients
b = [2,3,5,-3,0,4,0,8,-7,4]; a = 1;
[C,B,A] = dir2fs(b)

C =
    3.0867
   27.6302
    4.2979
   44.9952
   16.0000
  -16.0000
B =
    0.9719   -0.6479
    0.4152    0.7369
    0.6980   -0.4653
    0.0562   -0.5414
A =
    1.0000   -1.6180    1.0000
    1.0000   -0.6180    1.0000
    1.0000    0.6180    1.0000
    1.0000    1.6180    1.0000
    1.0000   -1.0000         0
    1.0000    1.0000         0
```

The block diagram is:



P6.19

Using the conjugate symmetry property of the DFT

$$H(k) = \begin{cases} H(0), & k = 0 \\ H^*(M-k), & k = 1, \ldots, M-1 \end{cases}$$

and the conjugate symmetry property of the *WM* -*k* factor, show that (6.12) can be put in the form (6.13) and (6.14) for real FIR filters.

Solutions

For real-valued FIR filters, the DFT, $\tilde{H}(k)$, is conjugate symmetric the DFT

$$\tilde{H}(k) = \begin{cases} \tilde{H}(0), & k = 0 \\ \tilde{H}^*(M-k) = \tilde{H}^*(-k), & k = 1, \ldots, M-1 \end{cases} \tag{6.9}$$

and the W_M^{-k} factor is also conjugate symmetric

$$W_M^{-k} = W_M^{M-k} = \left(W_M^k\right)^* \tag{6.10}$$

Then

$$H(z)=\left(\frac{1-z^{-M}}{M}\right)\sum_{k=0}^{M-1}\frac{\tilde{H}(k)}{1-W_M^{-k}z^{-1}} \tag{6.11}$$

can be put in the form

$$H(z)=\frac{1-z^{-M}}{M}\left\{\sum_{k=1}^{L}2\left|\tilde{H}(k)\right|H_k(z)+\frac{\tilde{H}(0)}{1-z^{-1}}+\frac{\tilde{H}(M/2)}{1+z^{-1}}\right\} \tag{6.12}$$

where $L=\frac{M-1}{2}$ for M odd, $L=\frac{M}{2}-1$ for M even, and

$$H_k(z)=\frac{\cos\left[\angle\tilde{H}(k)\right]-z^{-1}\cos\left[\angle\tilde{H}(k)-\frac{2\pi k}{M}\right]}{1-2z^{-1}\cos\left(\frac{2\pi k}{M}\right)+z^{-2}} \tag{6.13}$$

Proof. The sum in (6.11) can be expressed as (assuming M even)

$$\begin{aligned}
H(z)&=\left(\frac{1-z^{-M}}{M}\right)\left[\frac{\tilde{H}(0)}{1-z^{-1}}+\sum_{k=1}^{M/2-1}\frac{\tilde{H}(k)}{1-W_M^{-k}z^{-1}}+\frac{\tilde{H}(M/2)}{1+z^{-1}}+\sum_{k=M/2+1}^{M-1}\frac{\tilde{H}(k)}{1-W_M^{-k}z^{-1}}\right]\\
&=\left(\frac{1-z^{-M}}{M}\right)\left[\sum_{k=1}^{M/2-1}\frac{\tilde{H}(k)}{1-W_M^{-k}z^{-1}}+\sum_{k=M/2+1}^{M-1}\frac{\tilde{H}^*(M-k)}{1-\left(W_M^{k}\right)^*z^{-1}}+\frac{\tilde{H}(0)}{1-z^{-1}}+\frac{\tilde{H}(M/2)}{1+z^{-1}}\right]\\
&=\left(\frac{1-z^{-M}}{M}\right)\left[\sum_{k=1}^{M/2-1}\frac{\tilde{H}(k)}{1-W_M^{-k}z^{-1}}+\sum_{k=1}^{M/2-1}\frac{\tilde{H}^*(k)}{1-\left(W_M^{-k}\right)^*z^{-1}}+\frac{\tilde{H}(0)}{1-z^{-1}}+\frac{\tilde{H}(M/2)}{1+z^{-1}}\right]\\
&=\left(\frac{1-z^{-M}}{M}\right)\left[\sum_{k=1}^{M/2-1}\frac{\tilde{H}(k)-\tilde{H}(k)\left(W_M^{-k}\right)^*z^{-1}+\tilde{H}^*(k)-\tilde{H}^*(k)W_M^{-k}z^{-1}}{[1-W_M^{-k}z^{-1}][1-\left(W_M^{-k}\right)^*z^{-1}]}+\frac{\tilde{H}(0)}{1-z^{-1}}+\frac{\tilde{H}(M/2)}{1+z^{-1}}\right]
\end{aligned}$$

Consider

$$\begin{aligned}
&\frac{\tilde{H}(k)-\tilde{H}(k)\left(W_M^{-k}\right)^*z^{-1}+\tilde{H}^*(k)-\tilde{H}^*(k)W_M^{-k}z^{-1}}{[1-W_M^{-k}z^{-1}][1-\left(W_M^{-k}\right)^*z^{-1}]}\\
&=\frac{\tilde{H}(k)+\tilde{H}^*(k)-\tilde{H}(k)\left(W_M^{-k}\right)^*z^{-1}-\tilde{H}^*(k)W_M^{-k}z^{-1}}{1-2z^{-1}[W_M^{-k}+\left(W_M^{-k}\right)^*]+z^{-2}}\\
&=\frac{2\left|\tilde{H}(k)\right|\cos\left[\angle\tilde{H}\,(k)\right]-z^{-1}2\left|\tilde{H}(k)\right|\cos\left[\angle\tilde{H}(k)-\frac{2\pi k}{M}\right]}{1-2z^{-1}\cos\left(\frac{2\pi k}{M}\right)+z^{-2}}\\
&=2\left|\tilde{H}(k)\right|H_k(z)
\end{aligned}$$

which completes the proof.

P6.20

To avoid poles on the unit circle in the frequency sampling structure, one samples $H(z)$ at $z_k = re^{j2\pi/M}$, $k = 0, \ldots, M-1$ where $r \approx 1$(but <1), as discussed in Section 6.3.

1. Using

$$H\left(re^{j2\pi k/M}\right)\approx H(k),$$

show that the frequency-sampling structure is given by

$$H(z) = \frac{1-(rz)^{-M}}{M}\left\{\sum_{k=1}^{L} 2\left|H(k)\right| H_k(z) + \frac{H(0)}{1-rz^{-1}} + \frac{H(M/2)}{1+rz^{-1}}\right\}$$

where

$$H_k(z) = \frac{\cos\left[\angle H(k)\right] - rz^{-1}\cos\left[\angle H(k) - \frac{2\pi k}{M}\right]}{1-2rz^{-1}\cos\left(\frac{2\pi k}{M}\right)+r^2z^{-2}}, \quad k = 1,\ldots,L$$

and M is even.

2. Modify the MATLAB function **dir2fs** (which was developed in Section 6.3) to implement this frequency-sampling form. The format of this function should be

```
[C,B,A,rM] = dir2fs(h,r)
% Direct form to Frequency Sampling form conversion
% -------------------------------------------------
% [C,B,A,rM] = dir2fs(h,r)
% C = Row vector containing gains for parallel sections
% B = Matrix containing numerator coefficients arranged in rows
% A = Matrix containing denominator coefficients arranged in rows
% rM = r^M factor needed in the feedforward loop
% h = impulse response vector of an FIR filter
% r = radius of the circle over which samples are taken (r<1)
%
```

3. Determine the frequency sampling structure for the impulse response given in Example 6.6 using this function.

Solutions

1. Revised frequency-sampling structure: Replacing

$$z \to rz, \quad W_M^{-k} \to rW_M^{-k}, \quad \text{and} \quad \tilde{H}(k) \to H\left(re^{j2\pi k/M}\right) \approx \tilde{H}(k)$$

in (6.11), we obtain

$$\begin{aligned} H(z) \approx H(rz) &\approx \left(\frac{1-(rz)^{-M}}{M}\right)\sum_{k=0}^{M-1}\frac{\tilde{H}(k)}{1-rW_M^{-k}z^{-1}} \\ &= \left(\frac{1-(rz)^{-M}}{M}\right)\sum_{k=0}^{M-1}\frac{\tilde{H}(k)}{1-W_M^{-k}\left(rz^{-1}\right)} \end{aligned} \qquad (6.14)$$

Following steps similar to those in P6.19, we obtain

$$H(z) = \frac{1-(rz)^{-M}}{M}\left\{\sum_{k=1}^{L} 2\left|\tilde{H}(k)\right| H_k(z) + \frac{\tilde{H}(0)}{1-rz^{-1}} + \frac{\tilde{H}(M/2)}{1+rz^{-1}}\right\}$$

where

$$H_k(z) = \frac{\cos\left[\angle\tilde{H}(k)\right] - rz^{-1}\cos\left[\angle\tilde{H}(k) - \frac{2\pi k}{M}\right]}{1 - 2rz^{-1}\cos\left(\frac{2\pi k}{M}\right) + r^2 z^{-2}}, \quad k = 1, \ldots, L$$

and M is even.

2. New Matlab function **dir2fs**:

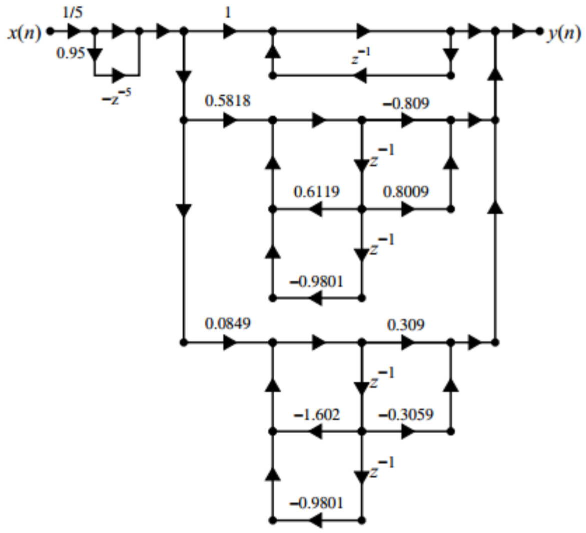
```
function [C,B,A,rM] = dir2fs(h,r)
% Direct form to Frequency Sampling form conversion
% -------------------------------------------------
% [C,B,A,rM] = dir2fs(h,r)
% C = Row vector containing gains for parallel sections
% B = Matrix containing numerator coefficients arranged
in rows
% A = Matrix containing denominator coefficients arranged
in rows
% rM = r^M factor needed in the feedforward loop
% h = impulse response vector of an FIR filter
% r = radius of the circle over which samples are taken
(r<1)
%
M = length(h);
if nargin == 1
r = 1; rM = 1;
elseif nargin == 2
rM = r^M;
end
H = fft(h,M); magH = abs(H); phaH = angle(H)';
% check even or odd M
if (M == 2*floor(M/2))
L = M/2-1; % M is even2006 Solutions Manual for DSP using
Matlab (2nd Edition) 309
A1 = [1,-r,0;1,r,0];
C1 = [real(H(1)),real(H(L+2))];
else
L = (M-1)/2; % M is odd
A1 = [1,-r,0];
C1 = [real(H(1))];
end
k = [1:L]';
% initialize B and A arrays
B = zeros(L,2); A = ones(L,3);
% compute denominator coefficients
A(1:L,2) = [-2*r*cos(2*pi*k/M)]; A(1:L,3) = [r*r]; A =
[A;A1];
% compute numerator coefficients
```

```
B(1:L,1) = cos(phaH(2:L+1));
B(1:L,2) = -r*cos(phaH(2:L+1)-(2*pi*k/M));
% compute gain coefficients
C = [2*magH(2:L+1),C1]';
```

3. The frequency sampling structure for the impulse response given in Example 6.6 using the above function: Matlab script:

```
% P6.20
%% P0620c.m
% 3. Example 6.6 impulse response
h = [1,2,3,2,1]/9;
[C,B,A,rM] = dir2fs(h,0.99)

C =
    0.5818
    0.0849
    1.0000
B =
   -0.8090    0.8009
    0.3090   -0.3059
A =
    1.0000   -0.6119    0.9801
    1.0000    1.6019    0.9801
    1.0000   -0.9900         0
rM =
    0.9510
```

Block diagram:



P6.21

Determine the impulse response of an FIR filter with lattice parameters

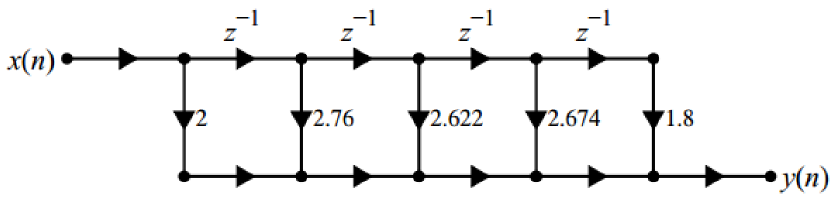
$$K_0 = 2, \quad K_1 = 0.6, \quad K_2 = 0.3, \quad K_3 = 0.5, \quad K_4 = 0.9$$

Draw the direct form and lattice form structures of this filter.

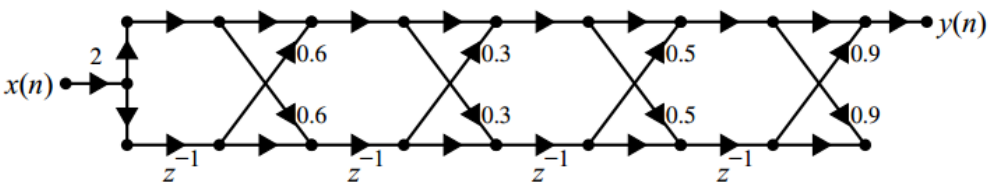
Solutions

```
% P6.21
% Direct form
% Given Lattice Structure
K = [2,0.6,0.3,0.5,0.9];
b = latc2dir(K)

b =
    2.0000    2.7600    2.6220    2.6740    1.8000
```

1. The block diagram of the direct form is



2. The block diagram of the lattice form is



P6.22

Consider the following system function of an FIR filter

$$H(z) = 1 - 4z^{-1} + 6.4z^{-2} - 5.12z^{-3} + 2.048z^{-4} - 0.32768z^{-5}$$

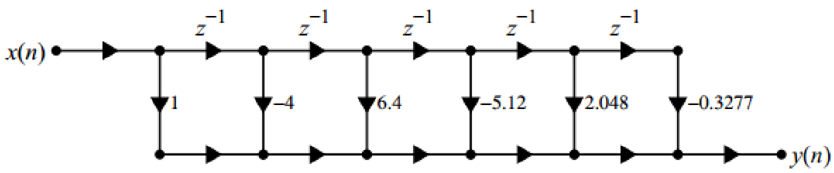
1. Provide block diagram structures in the following forms:
(a) Normal and transposed direct forms
(b) Cascade of five 1st-order sections
(c) Cascade of one 1st-order section and two 2nd-order sections
(d) Cascade of one 2nd-order section and one 3rd-order section
(e) Frequency-sampling structure with real coefficients
2. The computational complexity of a digital filter structure can be given by the total number of multiplications and the total number of 2-input additions that are required per output point.

Assume that $x(n)$ is real and that multiplication by 1 is not counted as a multiplication. Compare the computational complexity of each of these structures.

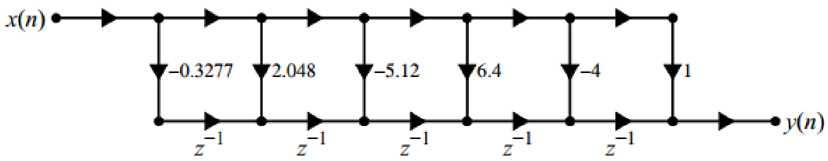
Solutions

1. Block diagram structures in the following form:

(a) Normal direct form:



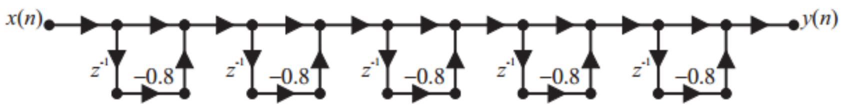
Transposed direct form:



(b) Cascade of five first-order sections: Matlab script:

```
%% P0622b.m
% Cascade of five first-order sections
b = [1,-4,6.4,-5.12,2.048,-0.32768];
broots = round(real(roots(b))*10)/10; broots = broots'

broots =
    0.8000    0.8000    0.8000    0.8000    0.8000
```

Block diagram:

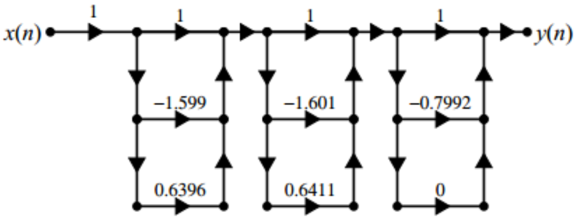


(c) Cascade of one first-order section and two second-order sections: Matlab script:

```
%% P0622c.m
%  Cascade of one first-order section and two second-order
sections:
% Given FIR filter
b = [1,-4,6.4,-5.12,2.048,-0.32768]; a = 1;
[b0,B,A] = dir2cas(b,a)

b0 =
     1
B =
    1.0000   -1.5995    0.6396
    1.0000   -1.6013    0.6411
```

```
    1.0000   -0.7992         0
A =
     1     0     0
     1     0     0
     1     0     0
```

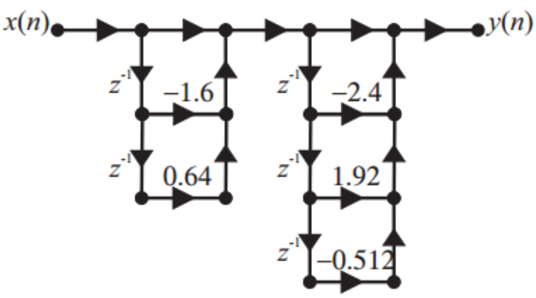
Block diagram:



(d) Cascade of one second-order section and one third-order section: Matlab script:

```
%% P0622d.m
% Cascade of one second-order section and one third-order
section:
% Given FIR filter
b = [1,-4,6.4,-5.12,2.048,-0.32768];
broots = round(real(roots(b))*10)/10; broots = broots';
b1 = poly(broots(1:2)), b2 = poly(broots(3:5))

b1 =
    1.0000   -1.6000    0.6400
b2 =
    1.0000   -2.4000    1.9200   -0.5120
```

Block diagram:

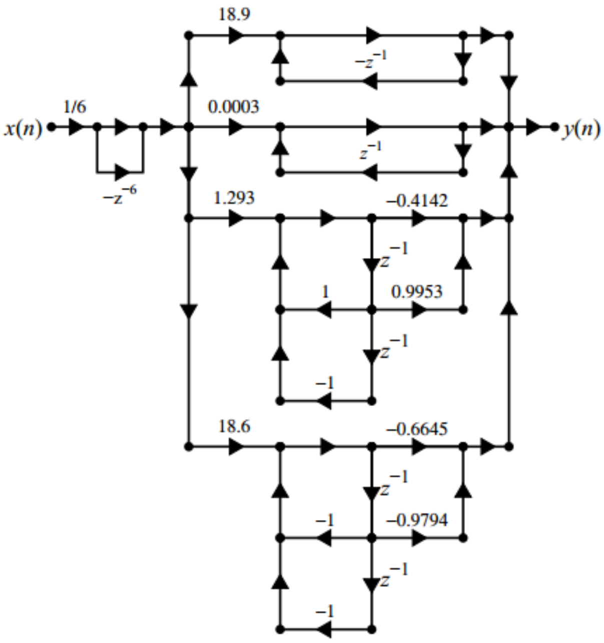


(e) Frequency-sampling structure with real coefficients: Matlab script:

```
%% P0622e.m
% Frequency-sampling structure with real coefficients:
% Given FIR filter
b = [1,-4,6.4,-5.12,2.048,-0.32768];
[C,B,A] = dir2fs(b)

C =
    1.2934
   18.5996
```

```
    0.0003
   18.8957
B =
   -0.4142    0.9953
   -0.6645   -0.9794
A =
    1.0000   -1.0000    1.0000
    1.0000    1.0000    1.0000
    1.0000   -1.0000         0
    1.0000    1.0000         0
```

Block diagram:



2. The computational complexity:

Structure	# of Mults	# of Adds
i. Direct	5	5
ii. Cascade-1	5	5
iii. Cascade-2	5	5
iv. Cascade-3	5	5
v. Freq. Samp	9	12

P6.23

A causal digital filter is described by the following zeros:

$$z_1 = 0.5\,e^{j60^\circ}, \quad z_2 = 0.5\,e^{-j60^\circ}, \quad z_3 = 2\,e^{j60^\circ}, \quad z_4 = 2\,e^{-j60^\circ},$$
$$z_5 = 0.25\,e^{j30^\circ}, \quad z_6 = 0.25\,e^{-j30^\circ}, \quad z_7 = 4\,e^{j30^\circ}, \quad z_8 = 4\,e^{-j30^\circ},$$

and poles: $\{p_i\}_{i=1}^{8} = 0$.

1. Determine the phase response of this filter, and show that it is a linear-phase FIR filter.
2. Determine the impulse response of the filter.

3. Draw a block diagram of the filter structure in the direct form.
4. Draw a block diagram of the filter structure in the linear-phase form.

Solutions

1. Phase response of the filter:

```
% P6.23
% Pole-zero description
r1 = 0.5; theta1 = 30; r2 = 0.25; theta2 = 60;
z1 = r1*exp( j*theta1*pi/180); z2 = r1*exp(-
j*theta1*pi/180);
z3 = 1/r1*exp( j*theta1*pi/180); z4 = 1/r1*exp(-
j*theta1*pi/180);
z5 = r2*exp( j*theta2*pi/180); z6 = r2*exp(-
j*theta2*pi/180);
z7 = 1/r2*exp( j*theta2*pi/180); z8 = 1/r2*exp(-
j*theta2*pi/180);
z = [z1;z2;z3;z4;z5;z6;z7;z8]; b = poly(z)

b =
   1.0000   -8.5801   42.7155 -113.2754  162.5092 -
113.2754   42.7155   -8.5801   1.0000
```

Hence

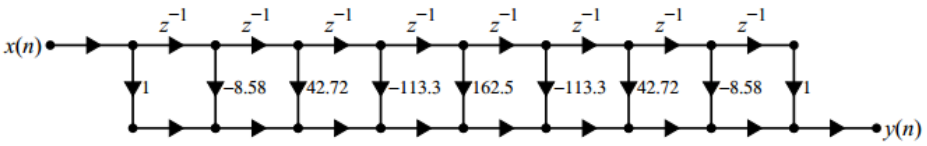
$$H(z) = 1 - 8.5801z^{-1} + 42.7155z^{-2} - 113.2754z^{-3} + 162.5092z^{-4} - 113.2754z^{-5} + 42.7155z^{-6} - 8.5801z^{-7} + z^{-8}$$

Due to symmetry in the coefficients, the phase response is $\angle H(e^{jw})$ = -4ω, which is linear.

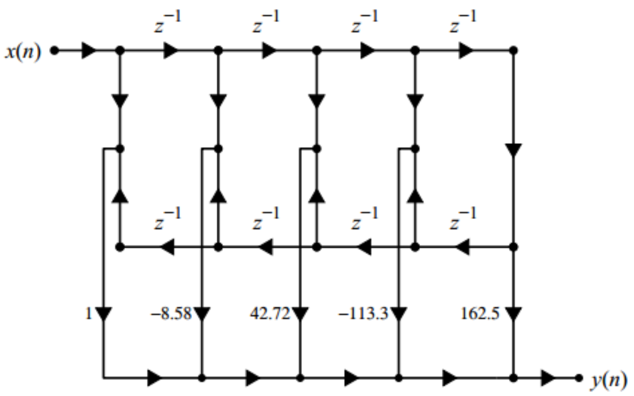
2. Impulse response of the filter:

$$h(n) = \{1, -8.5801, 42.7155, -113.2754, 162.5092, -113.2754, 42.7155, -8.5801, 1\}$$

3. Direct form structure:



4. Linear-phase form structure:



Chapter 7

P7.1

The absolute and relative (dB) specifications for a lowpass filter are related by (7.1) and (7.2). In this problem we will develop a simple MATLAB function to convert one set of specifications into another.

1. Write a MATLAB function to convert absolute specifications δ_1 and δ_2 into the relative specifications R_p and A_s in dB. The format of the function should be

function [Rp,As] = delta2db(delta1,delta2)
% Converts absolute specs delta1 and delta2 into dB specs Rp and As
% [Rp,As] = delta2db(delta1,delta2)

Verify your function using the specifications given in Example 7.2.

2. Write a MATLAB function to convert relative (dB) specifications R_p and A_s into the absolute specifications δ_1 and δ_2. The format of the function should be

function [delta1,delta2] = db2delta(Rp,As)
% Converts dB specs Rp and As into absolute specs delta1 and delta2
% [delta1,delta2] = db2delta(Rp,As)

Verify your function using the specifications given in Example 7.1.

Solutions

```
function [Rp,As] = delta2db(delta1,delta2)
% Conversion from Absolute delta specs to Relative dB
specs
% [Rp,As] = delta2db(delta1,delta2)
% Rp = Passband ripple
% As = Stopband attenuation
% d1 = Passband tolerance
% d2 = Stopband tolerance
Rp = -20*log10((1-delta1)/(1+delta1));
As = -20*log10(delta2/(1+delta1));
```

Matlab Verification:

```
% P7.1
clear;clc;
% Matlab Verification: using specifiations given in
example 7.2
```

```
delta1 = 0.01; delta2 = 0.001;
[Rp,As] = delta2db(delta1,delta2)

Rp =
    0.1737
As =
   60.0864

function [d1,d2] = db2delta(Rp,As)
% Conversion from Relative dB specs to Absolute delta
specs.
% [d1,d2] = db2delta(Rp,As)
% d1 = Passband tolerance
% d2 = Stopband tolerance
% Rp = Passband ripple
% As = Stopband attenuation
K = 10^(Rp/20);
d1 = (K-1)/(K+1); d2 = (1+d1)*(10^(-As/20));

% Matlab Verification: using specifiations given in
example 7.1
Rp = 0.25;As = 50;
[delta1,delta2] = db2delta(Rp,As)

delta1 =
    0.0144
delta2 =
    0.0032
```

P7.2

The Type-1 linear-phase FIR filter is characterized by

$$h(n) = h(M-1-n)), \quad 0 \le n \le M-1, \quad M \text{ odd}$$

Show that its amplitude response $H_r(\omega)$ is given by

$$H_r(\omega) = \sum_{n=0}^{L} a(n)\cos(\omega n), \quad L = \frac{M-1}{2}$$

where coefficients $\{a(n)\}$ are obtained as defined in (7.6).

Solutions

The Type-1 linear-phase FIR filter is characterized by a symmetric $h(n)$ and M-odd. Hence

$$h(n)=h(M-1-n)), \quad 0\le n\le M-1, \quad \alpha=\frac{M-1}{2} \text{ is an integer}$$

Consider the frequency response $H(e^{jw})$ given by

$$H(e^{j\omega})=\sum_{n=0}^{M-1}h(n)e^{-j\omega n}=\sum_{n=0}^{\frac{M-3}{2}}h(n)e^{-j\omega n}+h\left(\frac{M-1}{2}\right)e^{-j\omega\left(\frac{M-1}{2}\right)}+\sum_{n=\frac{M+1}{2}}^{M-1}h(n)e^{-j\omega n}$$

Using change of variables in the third sum:

$$n\to M-1-n\Rightarrow\frac{M+1}{2}\to\frac{M-3}{2}, \quad M-1\to 0, \text{ and } h(M-1-n)\to h(n)$$

we obtain

$$\begin{aligned}
H(e^{j\omega})&=\sum_{n=0}^{\frac{M-3}{2}}h(n)e^{-j\omega n}+h\left(\frac{M-1}{2}\right)e^{-j\omega\left(\frac{M-1}{2}\right)}+\sum_{n=0}^{\frac{M-3}{2}}h(n)e^{-j\omega(M-1-n)}\\
&=e^{-j\omega\left(\frac{M-1}{2}\right)}\left\{\sum_{n=0}^{\frac{M-3}{2}}h(n)e^{-j\omega n+j\omega\left(\frac{M-1}{2}\right)}+h\left(\frac{M-1}{2}\right)+\sum_{n=0}^{\frac{M-3}{2}}h(n)e^{-j\omega(M-1-n)+j\omega\left(\frac{M-1}{2}\right)}\right\}\\
&=e^{-j\omega\left(\frac{M-1}{2}\right)}\left\{h\left(\frac{M-1}{2}\right)+\sum_{n=0}^{\frac{M-3}{2}}h(n)\left[e^{+j\omega\left(\frac{M-1}{2}-n\right)}+e^{-j\omega\left(\frac{M-1}{2}-n\right)}\right]\right\}\\
&=e^{-j\omega\left(\frac{M-1}{2}\right)}\left\{h\left(\frac{M-1}{2}\right)+\sum_{n=0}^{\frac{M-3}{2}}2h(n)\cos\left[\omega\left(\frac{M-1}{2}-n\right)\right]\right\}
\end{aligned}$$

Hence

$$H_r(\omega)=\left\{h\left(\frac{M-1}{2}\right)+\sum_{n=0}^{\frac{M-3}{2}}2h(n)\cos\left[\omega\left(\frac{M-1}{2}-n\right)\right]\right\}$$

Define $a(n)=2h(\frac{M-1}{2}-n)$, $n=1,2,\ldots,(M-1)/2$ and $a(0)=h[(M-1)/2]$. Then

$$\begin{aligned}
H_r(\omega)&=\left\{a(0)+\sum_{n=1}^{\frac{M-1}{2}}a(n)\cos(\omega n)\right\}=\sum_{n=0}^{\frac{M-1}{2}}a(n)\cos(\omega n)\\
&=\sum_{n=0}^{L}a(n)\cos(\omega n); \quad L=\frac{M-1}{2}
\end{aligned}$$

P7.3

The Type-2 linear-phase FIR filter is characterized by

$$h(n)=h(M-1-n), \quad 0\le n\le M-1, \quad M \text{ even}$$

1. Show that its amplitude response $H_r(\omega)$ is given by

$$H_r(\omega)=\sum_{n=1}^{M/2}b(n)\cos\left\{\omega\left(n-\tfrac{1}{2}\right)\right\}$$

where coefficients $\{b(n)\}$ are obtained as defined in (7.10).

2. Show that $H_r(\omega)$ can be further expressed as

$$H_r(\omega) = \cos\left(\frac{\omega}{2}\right)\sum_{n=0}^{L}\tilde{b}(n)\cos(\omega n), \quad L = \frac{M}{2} - 1$$

where coefficients $\tilde{b}(n)$ are given by

$$\begin{aligned} b(1) &= \tilde{b}(0) + \tfrac{1}{2}\tilde{b}(1), \\ b(n) &= \frac{1}{2}\left[\tilde{b}(n-1) + \tilde{b}(n)\right], \quad 2 \le n \le \frac{M}{2} - 1, \\ b\left(\tfrac{M}{2}\right) &= \frac{1}{2}\tilde{b}\left(\tfrac{M}{2} - 1\right). \end{aligned}$$

Solutions

The Type-2 linear-phase FIR filter is characterized by symmetric $h(n)$ and M-even, i.e.,

$$h(n) = h(M-1-n), \; 0 \le n \le M-1; \quad \alpha = \frac{M-1}{2} \text{ is not an integer}$$

(a) Consider,

$$H\left(e^{j\omega}\right) = \sum_{n=0}^{M-1} h(n)\, e^{-j\omega n} = \sum_{n=0}^{\frac{M}{2}-1} h(n)\, e^{-j\omega n} + \sum_{\frac{M}{2}}^{M-1} h(n)\, e^{-j\omega n}$$

Change of variable in the second sum above:

$$n \to M-1-n \Rightarrow \frac{M}{2} \to \frac{M}{2} - 1, \; M-1 \to 0, \; \text{and } h(n) \to h(n)$$

Hence,

$$\begin{aligned} H\left(e^{j\omega}\right) &= \textstyle\sum_{n=0}^{\frac{M}{2}-1} h(n)\, e^{-j\omega n} + \sum_{n=0}^{\frac{M}{2}-1} h(n)\, e^{-j\omega(M-1-n)} \\ &= e^{-j\omega\left(\frac{M-1}{2}\right)} \textstyle\sum_{n=0}^{\frac{M}{2}-1} h(n) \left\{ e^{-j\omega n + j\omega\left(\frac{M-1}{2}\right)} + e^{-j\omega(M-1-n) + j\omega\left(\frac{M-1}{2}\right)} \right\} \\ &= e^{-j\omega\left(\frac{M-1}{2}\right)} \textstyle\sum_{n=0}^{\frac{M}{2}-1} h(n) \left\{ e^{+j\omega\left(\frac{M-1}{2}-n\right)} + e^{-j\omega\left(\frac{M-1}{2}-n\right)} \right\} \\ &= e^{-j\omega\left(\frac{M-1}{2}\right)} \textstyle\sum_{n=0}^{\frac{M}{2}-1} 2h(n) \cos\left[\left(\frac{M-1}{2} - n\right)\omega\right] \end{aligned}$$

Change of variable:

$$\frac{M}{2} - n \to n \Rightarrow n = 0 \to n = \frac{M}{2}, \; n = \frac{M}{2} - 1 \to n = 1$$

and

$$\cos\left[\left(\frac{M-1}{2} - n\right)\omega\right] \to \cos\left[\omega\left(n - \frac{1}{2}\right)\right]$$

Hence,

$$H\left(e^{j\omega}\right) = e^{-j\omega\left(\frac{M-1}{2}\right)} \sum_{n=1}^{\frac{M}{2}} 2h\left(\frac{M}{2} - n\right)\cos\left[\omega\left(n - \frac{1}{2}\right)\right]$$

Define $b(n) = 2h(\frac{M}{2} - n)$. Then,

$$H\left(e^{j\omega}\right) = e^{-j\omega\left(\frac{M-1}{2}\right)} \sum_{n=1}^{\frac{M}{2}} b\left(n\right)\cos\left[\omega\left(n-\frac{1}{2}\right)\right] \Rightarrow H_r\left(\omega\right) = \sum_{n=1}^{\frac{M}{2}} b\left(n\right)\cos\left[\omega\left(n-\frac{1}{2}\right)\right]$$

P7.4

The Type-3 linear-phase FIR filter is characterized by

$$h(n) = -h(M-1-n), \quad 0 \le n \le M-1, \quad M \text{ odd}$$

1. Show that its amplitude response $H_r(\omega)$ is given by

$$H_r(\omega) = \sum_{n=1}^{(M-1)/2} c(n)\sin(\omega n)$$

where coefficients $\{c(n)\}$ are obtained as defined in (7.13).
2. Show that $H_r(\omega)$ can be further expressed as

$$H_r(\omega) = \sin(\omega)\sum_{n=0}^{L} \tilde{c}(n)\cos(\omega n), \quad L = \frac{M-3}{2}$$

where coefficients $\tilde{c}(n)$ are given by

$$\begin{aligned} c(1) &= \tilde{c}(0) - \tfrac{1}{2}\tilde{c}(1), \\ c(n) &= \frac{1}{2}\left[\tilde{c}(n-1) - \tilde{c}(n)\right], \quad 2 \le n \le \frac{M-3}{2}, \\ c\left(\frac{M-1}{2}\right) &= \frac{1}{2}\tilde{c}\left(\frac{M-3}{2}\right). \end{aligned}$$

Solutions

The Type-3 linear-phase FIR filter is characterized by antisymmetric $h(n)$ and M odd, i.e.,

$$h(n) = -h(M-1-n),\ 0 \le n \le M-1;\ h\left(\frac{M-1}{2}\right) = 0; \quad \alpha = \frac{M-1}{2} \text{ is an integer}$$

(a) Consider,

$$H\left(e^{j\omega}\right) = \sum_{n=0}^{M-1} h\left(n\right)e^{-j\omega n} = \sum_{n=0}^{\frac{M-3}{2}} h\left(n\right)e^{-j\omega n} + \sum_{\frac{M+1}{2}}^{M-1} h\left(n\right)e^{-j\omega n}$$

Change of variable in the second sum above:

$$n \to M-1-n \Rightarrow \frac{M+1}{2} \to \frac{M-3}{2},\ M-1 \to 0,\ \text{and } h\left(n\right) \to -h\left(n\right)$$

Hence,

$$
\begin{aligned}
H\left(e^{j\omega}\right) &= \sum_{n=0}^{\frac{M-3}{2}} h\left(n\right) e^{-j\omega n} - \sum_{n=0}^{\frac{M-3}{2}} h\left(n\right) e^{-j\omega (M-1-n)} \\
&= e^{-j\left(\frac{M-1}{2}\right)} \sum_{n=0}^{\frac{M-3}{2}} h(n) \left\{ e^{-j\omega n + j\omega\left(\frac{M-1}{2}\right)} - e^{-j\omega(M-1-n)+j\omega\left(\frac{M-1}{2}\right)} \right\} \\
&= e^{-j\left(\frac{M-1}{2}\right)} \sum_{n=0}^{\frac{M-3}{2}} h(n) \left\{ e^{+j\omega\left(\frac{M-1}{2}-n\right)} - e^{-j\omega\left(\frac{M-1}{2}-n\right)} \right\} \\
&= je^{-j\left(\frac{M-1}{2}\right)} \sum_{n=0}^{\frac{M-3}{2}} 2h(n) \sin\left[\left(\frac{M-1}{2} - n\right)\omega\right]
\end{aligned}
$$

Change of variable:

$$\frac{M-1}{2} - n \rightarrow n \Rightarrow n = 0 \rightarrow n = \frac{M-1}{2},\ n = \frac{M-3}{2} \rightarrow n = 1$$

and

$$\sin\left[\left(\frac{M-1}{2} - n\right)\omega\right] \rightarrow \sin\left(\omega n\right)$$

Hence,

$$H\left(e^{j\omega}\right) = je^{-j\omega\left(\frac{M-1}{2}\right)} \sum_{n=1}^{\frac{M-1}{2}} 2h\left(\frac{M-1}{2} - n\right) \sin\left(\omega n\right)$$

Define $c(n) = 2h(\frac{M-1}{2} - n)$. Then,

$$H\left(e^{j\omega}\right) = je^{-j\omega\left(\frac{M-1}{2}\right)} \sum_{n=1}^{\frac{M-1}{2}} c\left(n\right) \sin\left(\omega n\right) \Rightarrow H_r\left(\omega\right) = \sum_{n=1}^{\frac{M-1}{2}} c\left(n\right) \sin\left(\omega n\right)$$

(b) Now $\sin(\omega n)$ can be expressed as $\sin(\omega)$ times a polynomial in $\cos(\omega)$ as

$$\sin(\omega n) = \sin(\omega) U_{n-1}[\cos(\omega)]$$

where U_n is a Chebyshev polynomial of the second kind. Thus $\sin(\omega n)$ can be written as a linear combination of higher harmonics in $\cos(\omega)$ multiplied by $\sin(\omega)$, i.e.,

$$
\begin{aligned}
\sin(\omega) &= \sin(\omega)\left\{\cos(0\omega)\right\} \\
\sin(2\omega) &= \sin(\omega)\left\{2\cos(\omega)\right\} \\
\sin(3\omega) &= \sin(\omega)\left\{\cos 0\omega + 2\cos 2\omega\right\}
\end{aligned}
$$

etc. Note that the lowest harmonic frequency is zero and the highest harmonic frequency is $(n - 1)\omega$ in the $\sin(\omega n)$ expansion. Hence,

$$H_r\left(\omega\right) = \sum_{n=1}^{\frac{M-1}{2}} c\left(n\right) \sin\left(\omega n\right) = \sin\omega \sum_{0}^{\frac{M-3}{2}} \tilde{c}\left(n\right) \cos\left(\omega n\right)$$

where $\tilde{c}(n)$ are related to $c(n)$ through the above trigonometric identities.

P7.5

The Type-4 linear-phase FIR filter is characterized by

$$h(n) = -h(M-1-n), \quad 0 \le n \le M-1, \quad M \text{ even}$$

1. Show that its amplitude response $H_r(\omega)$ is given by

$$H_r(\omega) = \sum_{n=1}^{M/2} d(n) \sin\left\{\omega\left(n - \tfrac{1}{2}\right)\right\}$$

where coefficients $\{d(n)\}$ are obtained as defined in (7.16).

2. Show that the above $H_r(\omega)$ can be further expressed as

$$H_r(\omega) = \sin\left(\frac{\omega}{2}\right) \sum_{n=0}^{L} \tilde{d}(n) \cos(\omega n), \quad L = \frac{M}{2} - 1$$

where coefficients $\tilde{d}(n)$ are given by

$$\begin{aligned} d(1) &= \tilde{d}(0) - \tfrac{1}{2}\tilde{d}(1), \\ d(n) &= \frac{1}{2}\left[\tilde{d}(n-1) - \tilde{d}(n)\right], \quad 2 \le n \le \frac{M}{2} - 1, \\ d\left(\frac{M}{2}\right) &= \frac{1}{2}\tilde{d}\left(\tfrac{M}{2} - 1\right). \end{aligned}$$

Solutions

The Type-4 linear-phase FIR filter is characterized by antisymmetric $h(n)$ and M-even, i.e.,

$$h(n) = -h(M-1-n), \; 0 \le n \le M-1; \quad \alpha = \frac{M-1}{2} \text{ is not an integer}$$

(a) Consider,

$$H\left(e^{j\omega}\right) = \sum_{n=0}^{M-1} h(n)\, e^{-j\omega n} = \sum_{n=0}^{\frac{M}{2}-1} h(n)\, e^{-j\omega n} + \sum_{\frac{M}{2}}^{M-1} h(n)\, e^{-j\omega n}$$

Change of variable in the second sum above:

$$n \to M-1-n \Rightarrow \frac{M}{2} \to \frac{M}{2} - 1, \; M-1 \to 0, \text{ and } h(M-1-n) \to -h(n)$$

Hence,

$$H\left(e^{j\omega}\right)=\sum_{n=0}^{\frac{M}{2}-1} h\left(n\right)e^{-j\omega n}-\sum_{n=0}^{\frac{M}{2}-1} h\left(n\right)e^{-j\omega(M-1-n)}$$

$$=e^{-j\omega\left(\frac{M-1}{2}\right)}\sum_{n=0}^{\frac{M}{2}-1} h(n)\left\{e^{-j\omega n+j\omega\left(\frac{M-1}{2}\right)}-e^{-j\omega(M-1-n)+j\omega\left(\frac{M-1}{2}\right)}\right\}$$

$$=e^{-j\omega\left(\frac{M-1}{2}\right)}\sum_{n=0}^{\frac{M}{2}-1} h(n)\left\{e^{+j\omega\left(\frac{M-1}{2}-n\right)}-e^{-j\omega\left(\frac{M-1}{2}-n\right)}\right\}$$

$$=e^{-j\omega\left(\frac{M-1}{2}\right)}\sum_{n=0}^{\frac{M}{2}-1} 2jh(n)\sin\left[\left(\frac{M-1}{2}-n\right)\omega\right]$$

Change of variable:

$$\frac{M}{2}-n\to n\Rightarrow n=0\to n=\frac{M}{2},\ n=\frac{M}{2}-1\to n=1$$

and

$$\sin\left[\left(\frac{M-1}{2}-n\right)\omega\right]\to\sin\left[\omega\left(n-\frac{1}{2}\right)\right]$$

Hence,

$$H\left(e^{j\omega}\right)=e^{-j\omega\left(\frac{\pi}{2}-\frac{M-1}{2}\right)}\sum_{n=1}^{\frac{M}{2}} 2h\left(\frac{M}{2}-n\right)\sin\left[\omega\left(n-\frac{1}{2}\right)\right]$$

Define $d(n) = 2h(\frac{M}{2} - n)$. Then,

$$H\left(e^{j\omega}\right)=e^{-j\omega\left(\frac{\pi}{2}\frac{M-1}{2}\right)}\sum_{n=1}^{\frac{M}{2}} d\left(n\right)\sin\left[\omega\left(n-\frac{1}{2}\right)\right]\Rightarrow H_r\left(\omega\right)=\sum_{n=1}^{\frac{M}{2}} d\left(n\right)\sin\left[\omega\left(n-\frac{1}{2}\right)\right]$$

(b) Now $\sin[\omega(n - \frac{1}{2})]$can be written as a linear combination of higher harmonics in $\cos\omega$ multiplied by $\sin(\frac{\omega}{2})$, i.e.,

$$\begin{aligned}\sin\left(\frac{1\omega}{2}\right) &= \sin\frac{\omega}{2}\{\cos 0\omega\}\\ \sin\left(\frac{3\omega}{2}\right) &= \sin\frac{\omega}{2}\{1+2\cos\omega\}\\ \sin\left(\frac{5\omega}{2}\right) &= \sin\frac{\omega}{2}\{\cos 0\omega+2\cos\omega+2\cos 2\omega\}\end{aligned}$$

etc. Note that the lowest harmonic frequency is zero and the highest harmonic frequency is $(n - 1)\omega$ in the $\sin\omega(n - \frac{1}{2})$ expansion. Hence,

$$H_r(\omega) = \sum_{n=1}^{\frac{M}{2}} d(n) \sin\left[\omega\left(n - \frac{1}{2}\right)\right] = \sin\frac{\omega}{2} \sum_{0}^{\frac{M}{2}-1} \tilde{d}(n) \cos(\omega n)$$

where $\tilde{d}(n)$ are related to $d(n)$ through the above trigonometric identities.

P7.6

Write a MATLAB function to compute the amplitude response $H_r(\omega)$ given a linear phase impulse response $h(n)$. The format of this function should be

function [Hr,w,P,L] = Ampl_Res(h);
% Computes Amplitude response Hr(w) and its polynomial P of order L,
% given a linear-phase FIR filter impulse response h.
% The type of filter is determined automatically by the subroutine.
%
% [Hr,w,P,L] = Ampl_Res(h)
% Hr = Amplitude Response
% w = frequencies between [0 pi] over which Hr is computed
% P = Polynomial coefficients
% L = Order of P
% h = Linear Phase filter impulse response

The function should first determine the type of the linear-phase FIR filter and then use the appropriate Hr_Type# function discussed in this chapter. It should also check if the given $h(n)$ is of a linear-phase type. Verify your function on sequences given here.

$$\begin{aligned}
h_{\mathrm{I}}(n) &= (0.9)^{|n-5|} \cos[\pi(n-5)/12]\,[u(n)-u(n-11)] \\
h_{\mathrm{II}}(n) &= (0.9)^{|n-4.5|} \cos[\pi(n-4.5)/11]\,[u(n)-u(n-10)] \\
h_{\mathrm{III}}(n) &= (0.9)^{|n-5|} \sin[\pi(n-5)/12]\,[u(n)-u(n-11)] \\
h_{\mathrm{IV}}(n) &= (0.9)^{|n-4.5|} \sin[\pi(n-4.5)/11]\,[u(n)-u(n-10)] \\
h(n) &= (0.9)^{n} \cos[\pi(n-5)/12]\,[u(n)-u(n-11)]
\end{aligned}$$

Solutions

Matlab Function:

```
function [Hr,w,P,L] = ampl_res(h);
%
% function [Hr,w,P,L] = ampl_res(h)
%
% Computes Amplitude response Hr(w) and its polynomial P
of order L,
% given a linear-phase FIR filter impulse response h.
```

```
% The type of filter is determined automatically by the
subroutine.
%
% Hr = Amplitude Response
% w = frequencies between [0 pi] over which Hr is
computed
% P = Polynomial coefficients
% L = Order of P
% h = Linear Phase filter impulse response
%
M = length(h);
L = floor(M/2);
if fix(abs(h(1:1:L))*10^10) ~= fix(abs(h(M:-1:M-
L+1))*10^10)
error('Not a linear-phase impulse response')
end
if 2*L ~= M
if fix(h(1:1:L)*10^10) == fix(h(M:-1:M-L+1)*10^10)
disp('*** Type-1 Linear-Phase Filter ***')
[Hr,w,P,L] = Hr_Type1(h);
elseif fix(h(1:1:L)*10^10) == -fix(h(M:-1:M-L+1)*10^10)
disp('*** Type-3 Linear-Phase Filter ***')
h(L+1) = 0;
[Hr,w,P,L] = Hr_Type3(h);
end
else
if fix(h(1:1:L)*10^10) == fix(h(M:-1:M-L+1)*10^10)
disp('*** Type-2 Linear-Phase Filter ***')
[Hr,w,P,L] = Hr_Type2(h);
elseif fix(h(1:1:L)*10^10) == -fix(h(M:-1:M-L+1)*10^10)
disp('*** Type-4 Linear-Phase Filter ***')
[Hr,w,P,L] = Hr_Type4(h);
end
end
```

Matlab Verification:

```
%% P7.6: Amplitude Response Function
clear;clc; close all;
%% 1. h_I(n)
n = 0:10; h_I = (0.9).^abs(n-5).*cos(pi*(n-5)/12);
[Hr,w,P,L] = ampl_res(h_I); P, L
% *** Type-1 Linear-Phase Filter ***

*** Type-1 Linear-Phase Filter ***
```

```
P =
   1.0000    1.7387    1.4030    1.0310    0.6561    0.3057
L =
    5
%% 2. h_II(n)
n = 0:9; h_II = (0.9).^abs(n-4.5).*cos(pi*(n-4.5)/11);
[Hr,w,P,L] = ampl_res(h_II); P, L
% *** Type-2 Linear-Phase Filter ***

*** Type-2 Linear-Phase Filter ***
P =
   1.8781    1.5533    1.1615    0.7478    0.3507
L =
    5
%% 3. h_III(n)
n = 0:10; h_III = (0.9).^abs(n-5).*sin(pi*(n-5)/12);
[Hr,w,P,L] = ampl_res(h_III); P, L
% *** Type-3 Linear-Phase Filter ***

*** Type-3 Linear-Phase Filter ***
P =
        0   -0.4659   -0.8100   -1.0310   -1.1364   -1.1407
L =
    5
%% 4. h_IV(n)
n = 0:9; h_IV = (0.9).^abs(n-4.5).*sin(pi*(n-4.5)/11);
[Hr,w,P,L] = ampl_res(h_IV); P, L
% *** Type-4 Linear-Phase Filter ***

*** Type-4 Linear-Phase Filter ***
P =
  -0.2700   -0.7094   -1.0064   -1.1636   -1.1944
L =
    5
%% 5. h_V(n)
n = 0:9; h_IV = (0.9).^n.*cos(pi*(n-5)/12);
[Hr,w,P,L] = ampl_res(h_IV); P, L
% ??? Error using ==> ampl_res

??? Error using ==>ampl_res (line 18)
Not a linear-phase impulse response
```

P7.7

Prove the following properties of linear-phase FIR filters

1. If $H(z)$ has four zeros at $z_1 = re^{j\theta}$, $z_2 = \frac{1}{r}e^{-j\theta}$, $z_3 = re^{-j\theta}$, and $z_4 = \frac{1}{r}e^{-j\theta}$ then $H(z)$ represents a linear-phase FIR filter.
2. If $H(z)$ has two zeros at $z_1 = e^{j\theta}$ and $z_2 = e^{-j\theta}$ then $H(z)$ represents a linear-phase FIR filter.
3. If $H(z)$ has two zeros at $z_1 = r$ and $z_2 = \frac{1}{r}$ then $H(z)$ represents a linear-phase FIR filter.
4. If $H(z)$ has a zero at $z_1 = 1$ or a zero at $z_1 = -1$ then $H(z)$ represents a linear-phase FIR filter.
5. For each of the sequences given in Problem P7.6, plot the locations of zeros. Determine which sequences imply linear-phase FIR filters.

Solutions

1. The filter $H(z)$ has the following four zeros

$$z_1 = re^{j\theta}, \quad z_2 = \frac{1}{r}e^{j\theta}, \quad z_3 = re^{-j\theta}, \quad z_4 = \frac{1}{r}e^{-j\theta}$$

The system function can be written as

$$\begin{aligned}
H(z) &= \left(1 - z_1 z^{-1}\right)\left(1 - z_2 z^{-1}\right)\left(1 - z_3 z^{-1}\right)\left(1 - z_4 z^{-1}\right) \\
&= \left(1 - re^{j\theta}z^{-1}\right)\left(1 - \frac{1}{r}e^{j\theta}z^{-1}\right)\left(1 - re^{-j\theta}z^{-1}\right)\left(1 - \frac{1}{r}e^{-j\theta}z^{-1}\right) \\
&= \left\{1 - (2r\cos\theta)\,z^{-1} + r^2 z^{-2}\right\}\left\{1 - \left(2r^{-1}\cos\theta\right)z^{-1} + r^{-2}z^{-2}\right\} \\
&= 1 - 2\cos\theta\left(r + r^{-1}\right)z^{-1} + \left(r^2 + r^{-2} + 4\cos^2\theta\right)z^{-2} - 2\cos\theta\left(r + r^{-1}\right)z^{-3} + z^{-4}
\end{aligned}$$

Hence the impulse response of the filter is

$$h(n) = \left\{\underset{\uparrow}{1}, -2\cos\theta\left(r + r^{-1}\right), \left(r^2 + r^{-2} + 4\cos^2\theta\right), -2\cos\theta\left(r + r^{-1}\right), 1\right\}$$

which is a finite-duration symmetric impulse response. This implies that the filter is a linear-phase FIR filter.

2. The filter $H(z)$ has the following two zeros

$$z_1 = e^{j\theta} \text{ and } \quad z_2 = e^{-j\theta}$$

The system function can be written as

$$\begin{aligned}
H(z) &= \left(1 - z_1 z^{-1}\right)\left(1 - z_2 z^{-1}\right) = \left(1 - e^{j\theta}z^{-1}\right)\left(1 - e^{-j\theta}z^{-1}\right) \\
&= \left\{1 - (2\cos\theta)\,z^{-1} + z^{-2}\right\}
\end{aligned}$$

Hence the impulse response of the filter is

$$h(n) = \left\{\underset{\uparrow}{1}, -2\cos\theta, 1\right\}$$

which is a finite-duration symmetric impulse response. This implies that the filter is a linear-phase FIR filter.

3. The filter $H(z)$ has the following two zeros

$$z_1 = r \text{ and } z_2 = \frac{1}{r}$$

The system function can be written as

$$H(z) = \left(1 - z_1 z^{-1}\right)\left(1 - z_2 z^{-1}\right) = \left(1 - r z^{-1}\right)\left(1 - \frac{1}{r} z^{-1}\right)$$
$$= 1 - \left(r + r^{-1}\right) z^{-1} + z^{-2}$$

Hence the impulse response of the filter is

$$h(n) = \left\{ \underset{\uparrow}{1}, -\left(r + r^{-1}\right), 1 \right\}$$

which is a finite-duration symmetric impulse response. This implies that the filter is a linear-phase FIR filter.

4. If $H(z)$ has a zero at $z_1 = 1$ or a zero at $z_1 = -1$ then $H(z)$ can be written as

$$H(z) = \left(1 - z^{-1}\right) \text{ or } H(z) = \left(1 + z^{-1}\right)$$

with impulse responses

$$h(n) = \left\{ \underset{\uparrow}{1}, 1 \right\} \text{ or } h(n) = \left\{ \underset{\uparrow}{1}, -1 \right\}$$

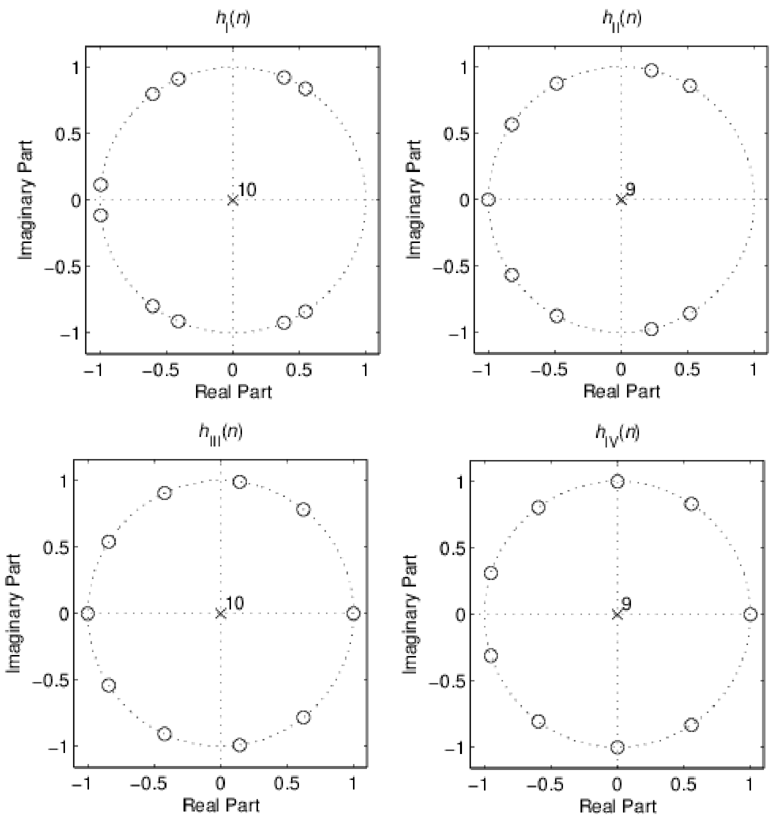
both of which are finite-duration with symmetric and antisymmetric impulse responses, respectively. This implies that the filter is a linear-phase FIR filter.

5. Zero plots using Matlab :

```
%% P7.7: Pole-Zero Plots of Linear-Phase Filters
clear;clc; close all;
%% 1. h_I(n)
n = 0:10; h_I = (0.9).^abs(n-5).*cos(pi*(n-5)/12);
Hf_1 = figure('Units','inches','position',[1,1,3,3],...
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,3,3]);
set(Hf_1,'NumberTitle','off','Name','P7.7.1');
zplane(h_I,1); title('{\ith}_{I}({\itn})')
%% 2. h_II(n)
n = 0:9; h_II = (0.9).^abs(n-4.5).*cos(pi*(n-4.5)/11);
Hf_2 = figure('Units','inches','position',[1,1,3,3],...
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,3,3]);
set(Hf_2,'NumberTitle','off','Name','P7.7.2');
zplane(h_II,1); title('{\ith}_{II}({\itn})')
%% 3. h_III(n)
n = 0:10; h_III = (0.9).^abs(n-5).*sin(pi*(n-5)/12);
```

```
Hf_3 = figure('Units','inches','position',[1,1,3,3],...
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,3,3]);
set(Hf_3,'NumberTitle','off','Name','P7.7.3');
zplane(h_III,1); title('{\ith}_{III}({\itn})')
%% 4. h_IV(n)
n = 0:9; h_IV = (0.9).^abs(n-4.5).*sin(pi*(n-4.5)/11);
Hf_4 = figure('Units','inches','position',[1,1,3,3],...
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,3,3]);
set(Hf_4,'NumberTitle','off','Name','P7.7.4');
zplane(h_IV,1); title('{\ith}_{IV}({\itn})')
%% 5. h(n)
n = 0:9; h = (0.9).^n.*cos(pi*(n-5)/12);
Hf_5 = figure('Units','inches','position',[1,1,3,3],...
'color',[0,0,0],'paperunits','inches','paperposition',[0,
0,3,3]);
set(Hf_5,'NumberTitle','off','Name','P7.7.5');
zplane(h,1); title('{\ith}({\itn})')
```

The zero-plots are shown in Figure 7.1. Clearly the first four plots satisfy the zero-placement requirements and hence the corresponding filters are linear-phase filters.



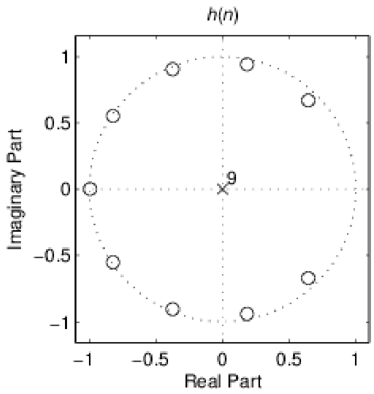


Figure 7.1: Plots of zeros in Problem 7.7

P7.8

A notch filter is an LTI system, which is used to eliminate an arbitrary frequency $\omega = \omega_0$. The ideal linear-phase notch filter frequency response is given by

$$H_d\left(e^{j\omega}\right) = \begin{cases} 0, & |\omega| = \omega_0; \\ 1 \cdot e^{-j\alpha\omega}, & \text{otherwise.} \end{cases} \qquad (\alpha \text{ is a delay in samples})$$

1. Determine the ideal impulse response, $h_d(n)$, of the ideal notch filter.
2. Using $h_d(n)$, design a linear-phase FIR notch filter using a length 51 rectangular window to eliminate the frequency $\omega_0 = \pi/2$ rad/sample. Plot amplitude the response of the resulting filter.
3. Repeat part 2 using a length 51 Hamming window. Compare your results

P7.9

Design a linear-phase bandpass filter using the Hann window design technique. The specifications are

lower stopband edge: 0.2π
upper stopband edge: 0.75π $A_s = 40$ dB
lower passband edge: 0.35π
upper passband edge: 0.55π $R_p = 0.25$ dB

Plot the impulse response and the magnitude response (in dB) of the designed filter. Do not use the **fir1** function.

Solutions

```
% P7.9
clear;clc; close all;
%% Specifications:
ws1 = 0.2*pi; % lower stopband edge
wp1 = 0.35*pi; % lower passband edge
wp2 = 0.55*pi; % upper passband edge
```

```
ws2 = 0.75*pi; % upper stopband edge
Rp = 0.25; % passband ripple
As = 40; % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = min((wp1-ws1),(ws2-wp2));
M = ceil(6.2*pi/tr_width); M = 2*floor(M/2)+1, % choose
odd M
n = 0:M-1; w_han = (hann(M))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(wc2,M)-ideal_lp(wc1,M); h = hd .* w_han;
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Rpd = -min(db((wp1/delta_w)+1:(wp2/delta_w)+1)), % Actual
passband ripple
Asd = floor(-max(db(1:(ws1/delta_w)+1))), % Actual Attn
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.9');
subplot(2,2,1); Hs_1= stem(n,hd,'filled');
set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)')
subplot(2,2,2); Hs_1 = stem(n,w_han,'filled');
set(Hs_1,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{han}(n)');
title('Hann Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8)
subplot(2,2,3); Hs_1 = stem(n,h,'filled');
set(Hs_1,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
```

```
ylabel('h(n)')
subplot(2,2,4); plot(w/pi,db,'linewidth',1);
title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels')
set(gca,'XTick',[0;0.2;0.35;0.55;0.75;1])
set(gca,'XTickLabel',{'0';'0.2';'0.35';'0.55';'0.75';'1'}
,'fontsize',8)
set(gca,'YTick',[-40;0]); set(gca,'YTickLabel',[' 40';' 0
']);grid
print -deps2 ../epsfiles/P0709
```

```
M =
    43
Rpd =
    0.1030
Asd =
    44
```

The filter response plots are shown in Figure 7.2.

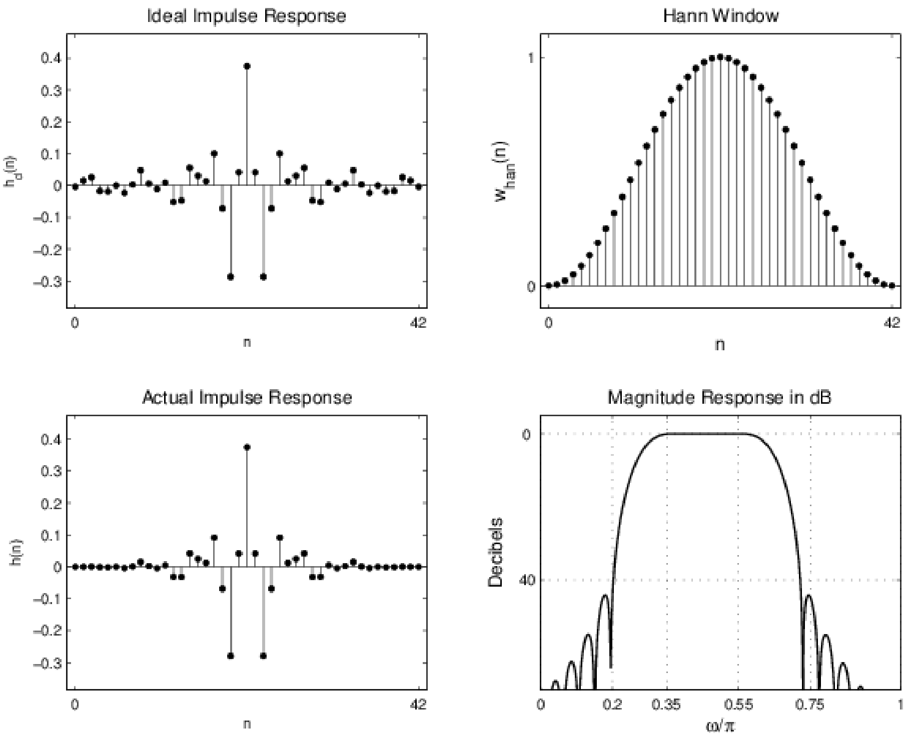


Figure 7.2: Filter design plots in Problem 7.9

P7.10

Design a bandstop filter using the Hamming window design technique. The specifications are

lower stopband edge: 0.4π

upper stopband edge: 0.6π $A_s = 50$ dB
lower passband edge: 0.3π
upper passband edge: 0.7π $R_p = 0.2$ dB

Plot the impulse response and the magnitude response (in dB) of the designed filter. Do not use the **fir1** function.

Solutions

```
% P7.10
clear;clc; close all;
%% Specifications:
wp1 = 0.3*pi; % lower passband edge
ws1 = 0.4*pi; % lower stopband edge
ws2 = 0.6*pi; % upper stopband edge
wp2 = 0.7*pi; % upper passband edge
Rp = 0.2; % passband ripple
As = 50; % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = abs(min((wp1-ws1),(ws2-wp2)));
M = ceil(6.6*pi/tr_width); M = 2*floor(M/2)+1, % choose
odd M
n = 0:M-1; w_ham = (hamming(M))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(pi,M)+ideal_lp(wc1,M)-ideal_lp(wc2,M); h =
hd .* w_ham;
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Asd = floor(-
max(db(ceil(ws1/delta_w)+1:floor(ws2/delta_w)+1))), %
Actual Attn
Rpd = -min(db(1:floor(wp1/delta_w)+1)), % Actual passband
ripple
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
```

```
set(Hf_1,'NumberTitle','off','Name','P7.10');
subplot(2,2,1); Hs_1= stem(n,hd,'filled');
set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)')
subplot(2,2,2); Hs_1 = stem(n,w_ham,'filled');
set(Hs_1,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Hamming Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8)
subplot(2,2,3); Hs_1 = stem(n,h,'filled');
set(Hs_1,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)')
subplot(2,2,4); plot(w/pi,db,'linewidth',1);
title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels')
set(gca,'XTick',[0;0.3;0.4;0.6;0.7;1])
set(gca,'XTickLabel',{'0';'0.3';'0.4';'0.6';'0.7';'1'},'f
ontsize',8)
set(gca,'YTick',[-50;0]);
set(gca,'YTickLabel',{'50';'0'});grid
print -deps2 ../EPSFILES/P0710

M =
    67
Asd =
    50
Rpd =

    0.0435
```

The filter response plots are shown in Figure 7.3.

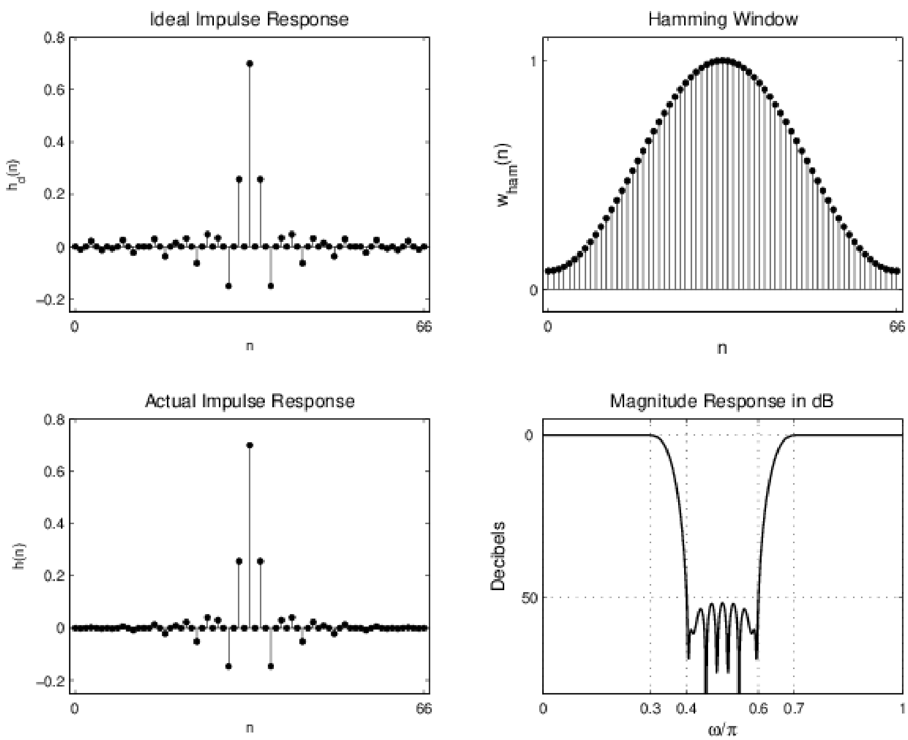


Figure 7.3: Filter design plots in Problem 7.10

P7.11

Design a bandpass filter using the Hamming window design technique. The specifications are

lower stopband edge: 0.3π

upper stopband edge: 0.6π $A_s = 50$ dB

lower passband edge: 0.4π

upper passband edge: 0.5π $R_p = 0.5$ dB

Plot the impulse response and the magnitude response (in dB) of the designed filter. Do not use the **fir1** function.

Solutions

```
% P7.11
clear;clc; close all;
%% Specifications:
ws1 = 0.3*pi; % lower stopband edge
wp1 = 0.4*pi; % lower passband edge
wp2 = 0.5*pi; % upper passband edge
ws2 = 0.6*pi; % upper stopband edge
Rp = 0.5; % passband ripple
As = 50; % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in
```

```
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = abs(min((wp1-ws1),(ws2-wp2)));
M = ceil(6.6*pi/tr_width); M = 2*floor(M/2)+1, % choose
odd M
n = 0:M-1; w_ham = (hamming(M))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(wc2,M)-ideal_lp(wc1,M); h = hd .* w_ham;
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Rpd = -
min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w)+1)), %
Actual passband ripple
Asd = floor(-max(db(1:(ws1/delta_w)+1))), % Actual Attn
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.11');
subplot(2,2,1); Hs_1= stem(n,hd,'filled');
set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)')
subplot(2,2,2); Hs_1 = stem(n,w_ham,'filled');
set(Hs_1,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Hamming Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8)
subplot(2,2,3); Hs_1 = stem(n,h,'filled');
set(Hs_1,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)')
subplot(2,2,4); plot(w/pi,db,'linewidth',1);
title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
```

```
ylabel('Decibels')
set(gca,'XTick',[0;0.3;0.4;0.5;0.6;1])
set(gca,'XTickLabel',{' 0 ';'0.3';'0.4';'0.5';'0.6';' 1
'},'fontsize',8)
set(gca,'YTick',[-50;0]); set(gca,'YTickLabel',{' 50';' 0
'});grid
print -deps2 ../EPSFILES/P0711

M =
    67
Rpd =
    0.0488
Asd =
    51
```

The filter response plots are shown in Figure 7.4.

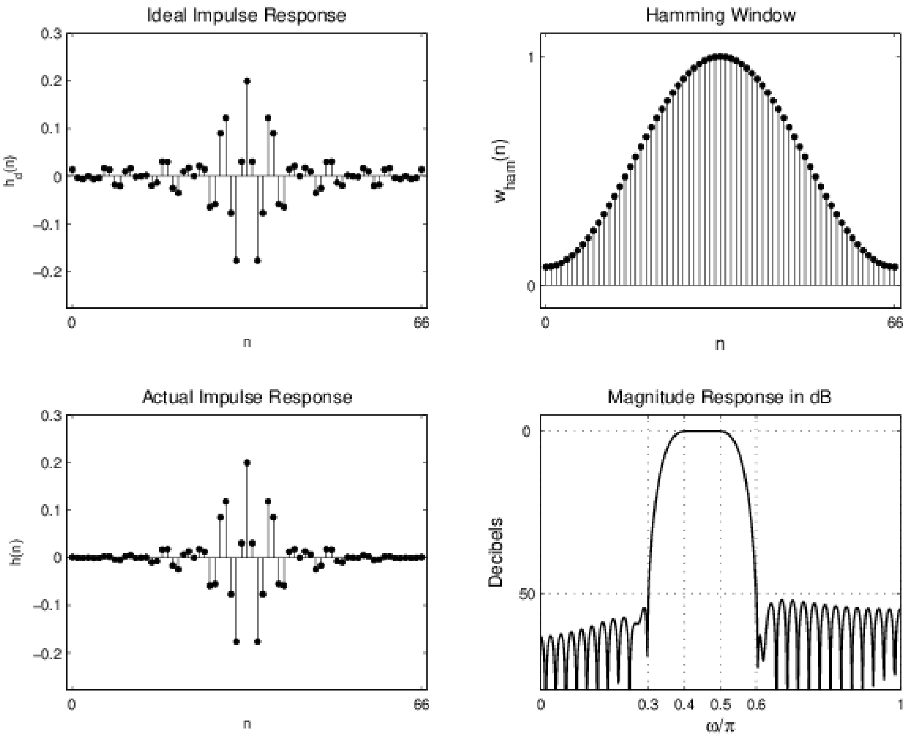


Figure 7.4: Filter design plots in Problem 7.11

P7.12

Design a highpass filter using one of the fixed window functions. The specifications are

stopband edge: 0.4π, $A_s = 50$ dB

passband edge: 0.6π, $R_p = 0.004$ dB

Plot the zoomed magnitude response (in dB) of the designed filter in the passband to verify the passband ripple R_p. Do not use the **fir1** function.

```
% P7.12
clear;clc; close all;
%% Specifications:
ws = 0.4*pi; % lower stopband edge
wp = 0.6*pi; % lower passband edge
Rp = 0.001; % passband ripple
As = 50; % stopband attenuation
%
%Select the min(delta1,delta2) since delta1=delta2 in
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = abs(wp-ws);
% using Blackman window
M = ceil(11*pi/tr_width); M = 2*floor(M/2)+1, % choose
odd M
n = 0:M-1; w_blk = (blackman(M))';
wc = (ws+wp)/2;
hd = ideal_lp(pi,M)-ideal_lp(wc,M); h = hd.*w_blk;
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Rpd = -min(db(ceil(wp/delta_w)+1:floor(pi/delta_w)+1)), %
Actual passband ripple
Asd = floor(-max(db(1:(ws/delta_w)+1))), % Actual Attn
%% Zoomed Filter Response Plot
Hf_1 = figure('Units','inches','position',[1,1,5,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.12');
plot(w(301:501)/pi,db(301:501),'linewidth',1);
title('Zoomed Magnitude Response in dB');
axis([0.6,1,-0.005,0.001]); xlabel('\omega/\pi');
ylabel('Decibels')
set(gca,'XTick',[0.6;1])
set(gca,'XTickLabel',{'0.6';' 1 '},'fontsize',8)
set(gca,'YTick',[-0.004;0]); set(gca,'YTickLabel',{'-
0.004';' 0 '});grid
print -deps2 ../EPSFILES/P0712
```

```
Delta1 is smaller than delta2
Rp =
   1.0000e-03
As =
   84.7974
M =
    55
Rpd =
    0.0039
Asd =
    71
```

The zoomed magnitude filter response plot is shown in Figure 7.5.

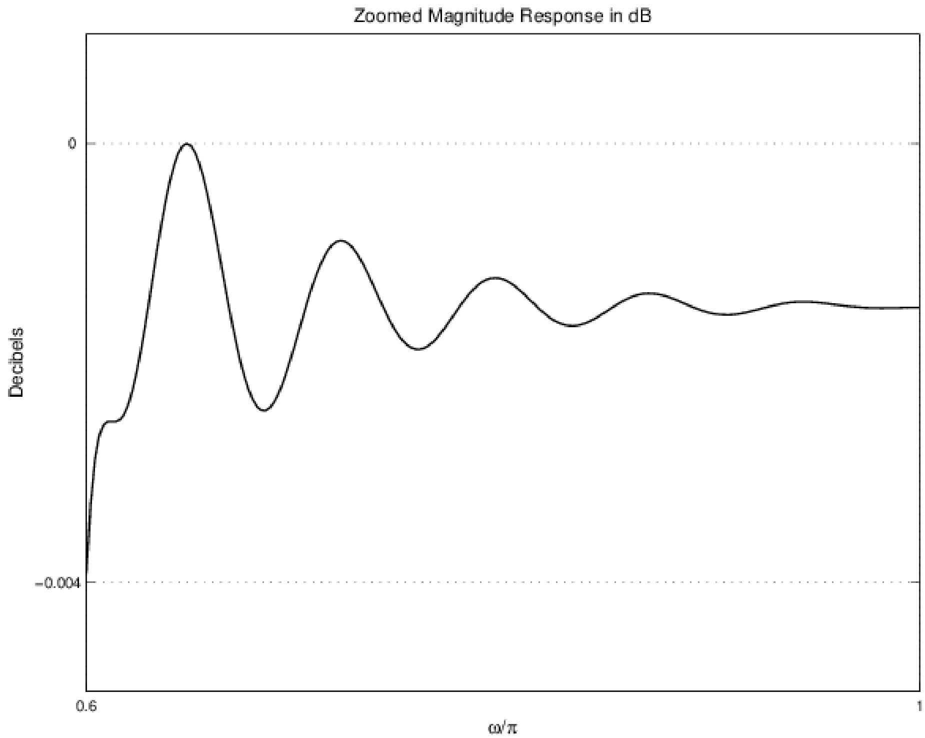


Figure 7.5: Filter design plots in Problem 7.12

P7.13

Using the Kaiser window method, design a linear-phase FIR digital filter that meets the following specifications

$$0.975 \le |H(e^{jw})| \le 1.025,\ 0 \le \omega \le 0.25\pi$$
$$0 \le |H(e^{jw})| \le 0.005,\ 0.35\pi \le \omega \le 0.65\pi$$
$$0.975 \le |H(e^{jw})| \le 1.025,\ 0.75\pi \le \omega \le \pi$$

Determine the minimum length impulse response $h(n)$ of such a filter. Provide a plot containing subplots of the amplitude response and the magnitude response in dB. Do not use the **fir1** function.

```
% P7.13
clear;clc; close all;
%% Specifications:
wp1 = 0.25*pi; % lower passband edge
ws1 = 0.35*pi; % lower stopband edge
ws2 = 0.65*pi; % upper stopband edge
wp2 = 0.75*pi; % upper passband edge
delta1 = 0.025; % passband ripple
delta2 = 0.005; % stopband ripple
%
% Convert to decibels
[Rp,As] = delta2db(delta1,delta2)
%
tr_width = abs(min((wp1-ws1),(ws2-wp2))); M = ceil((As-
7.95)/(2.285*tr_width)+1)+1;
M = 2*floor(M/2)+1, % choose odd M

n = [0:1:M-1]; beta = 0.1102*(As-8.7); w_kai =
(kaiser(M,beta))'; % Kaiser Window
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(pi,M)+ideal_lp(wc1,M)-ideal_lp(wc2,M); %
Ideal HP Filter
h = hd .* w_kai; % Window design
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Asd = floor(-
max(db(ceil(ws1/delta_w)+1:floor(ws2/delta_w)+1))), %
Actual Attn

Rpd = -min(db(1:floor(wp1/delta_w)+1)), % Actual passband
ripple

[Hr,w,P,L] = ampl_res(h);
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,7,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.13');
subplot('position',[0.08,0.6,0.25,0.35]);
Hs_1= stem(n,hd,'filled'); set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
```

```
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)');
subplot('position',[0.41,0.6,0.25,0.35]);
Hs_2 = stem(n,w_kai,'filled'); set(Hs_2,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Kaiser Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8);
subplot('position',[0.74,0.6,0.25,0.35]);
Hs_3 = stem(n,h,'filled'); set(Hs_3,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8);
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)');
subplot('position',[0.09,0.1,0.4,0.35]);
plot(w/pi,db,'linewidth',1); title('Magnitude Response in
dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels');
set(gca,'XTick',[0;0.25;0.35;0.65;0.75;1])
set(gca,'XTickLabel',{' 0 ';'0.25';'0.35';'0.65';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[-Asd;0]); set(gca,'YTickLabel',{' 49';'
0 '});grid;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,Hr,'linewidth',1); title('Apmlitude Response');
axis([0,1,-0.05,1.05]); xlabel('\omega/\pi');
ylabel('Amplitude');
set(gca,'XTick',[0;0.25;0.35;0.65;0.75;1]);
set(gca,'XTickLabel',{' 0 ';'0.25';'0.35';'0.65';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[0;1]); grid;
print -deps2 ../EPSFILES/P0713

Rp =
    0.4344
As =
   46.2351
M =
    57
Asd =
    49
Rpd =
    0.0492
```

```
*** Type-1 Linear-Phase Filter ***
```

The filter response plots are shown in Figure 7.6.

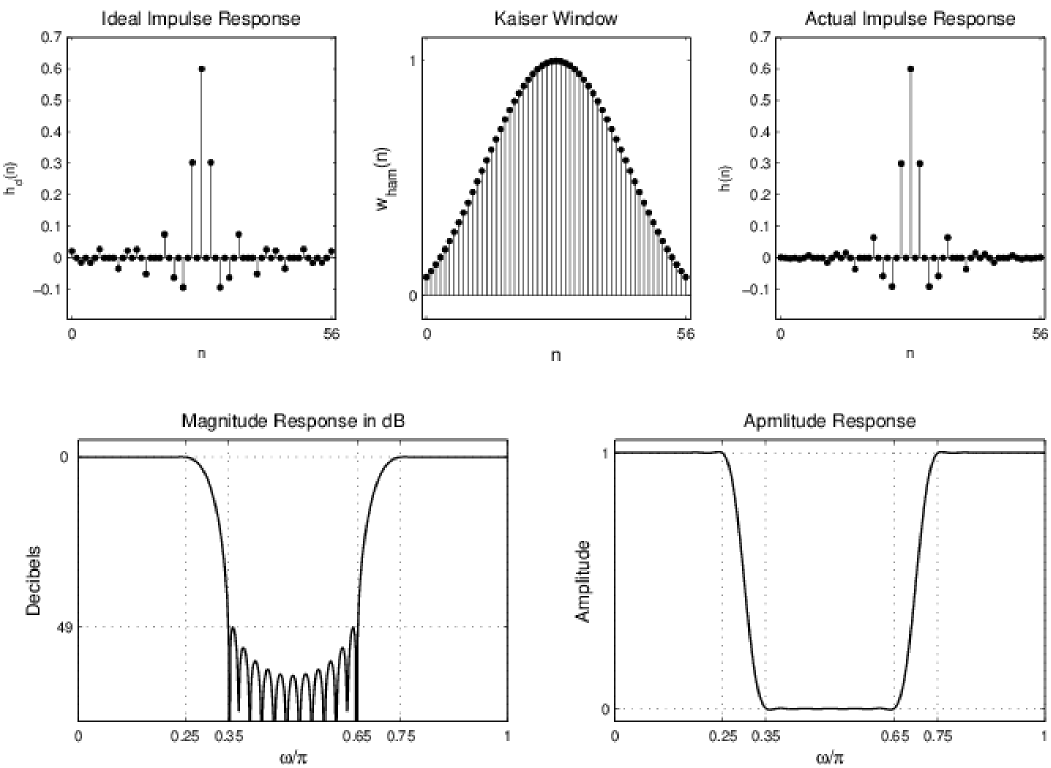


Figure 7.6: Filter design plots in Problem 7.13

P7.14

We wish to use the Kaiser window method to design a linear-phase FIR digital filter that meets the following specifications:

$$0 \leq |H(e^{jw})| \leq 0.01,\ 0 \leq \omega \leq 0.25\pi$$
$$0.95 \leq |H(e^{jw})| \leq 1.05,\ 0.35\pi \leq \omega \leq 0.65\pi$$
$$0 \leq |H(e^{jw})| \leq 0.01,\ 0.75\pi \leq \omega \leq \pi$$

Determine the minimum length impulse response $h(n)$ of such a filter. Provide a plot containing subplots of the amplitude response and the magnitude response in dB. Do not use the **fir1** function.

Solutions

```
% P7.14
clear;clc; close all;
%% Specifications:
ws1 = 0.25*pi; % lower stopband edge
wp1 = 0.35*pi; % lower passband edge
wp2 = 0.65*pi; % upper passband edge
ws2 = 0.75*pi; % upper stopband edge
```

```
delta1 = 0.05; % passband ripple
delta2 = 0.01; % stopband ripple
%
% Convert to decibels
[Rp,As] = delta2db(delta1,delta2)
%
tr_width = abs(min((wp1-ws1),(ws2-wp2)));
M = ceil((As-7.95)/(2.285*tr_width)+1)+1; M =
2*floor(M/2)+1, % choose odd M
n = [0:1:M-1]; beta = 0.1102*(As-8.7); w_kai =
(kaiser(M,beta))'; % Kaiser Window
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(wc2,M)-ideal_lp(wc1,M);
% Determine the Window Design Impulse Response and
Frequency Response
h = hd .* w_kai; [db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Rpd = -min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w))), %
Actual passband ripple
Asd = floor(-max(db(1:floor(ws1/delta_w)+1))), % Actual
Attn
[Hr,w,P,L] = ampl_res(h);
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,7,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.14');
subplot('position',[0.08,0.6,0.25,0.35]);
Hs_1= stem(n,hd,'filled'); set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)');
subplot('position',[0.41,0.6,0.25,0.35]);
Hs_2 = stem(n,w_kai,'filled'); set(Hs_2,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Kaiser Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8);
subplot('position',[0.74,0.6,0.25,0.35]);
Hs_3 = stem(n,h,'filled'); set(Hs_3,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8);
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
```

```
ylabel('h(n)');
subplot('position',[0.09,0.1,0.4,0.35]);
plot(w/pi,db,'linewidth',1); title('Magnitude Response in
dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels');
set(gca,'XTick',[0;0.25;0.35;0.65;0.75;1])
set(gca,'XTickLabel',{' 0 ';'0.25';'0.35';'0.65';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[-Asd;0]); set(gca,'YTickLabel',{' 42';'
0 '});grid;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,Hr,'linewidth',1); title('Apmlitude Response');
axis([0,1,-0.05,1.05]); xlabel('\omega/\pi');
ylabel('Amplitude');
set(gca,'XTick',[0;0.25;0.35;0.65;0.75;1]);
set(gca,'XTickLabel',{' 0 ';'0.25';'0.35';'0.65';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[0;1]); grid;
print -deps2 ../EPSFILES/P0714

Rp =
    0.8693
As =
   40.4238
M =
    49
Rpd =
    0.1033
Asd =
    42
*** Type-1 Linear-Phase Filter ***
```

The filter response plots are shown in Figure 7.7.

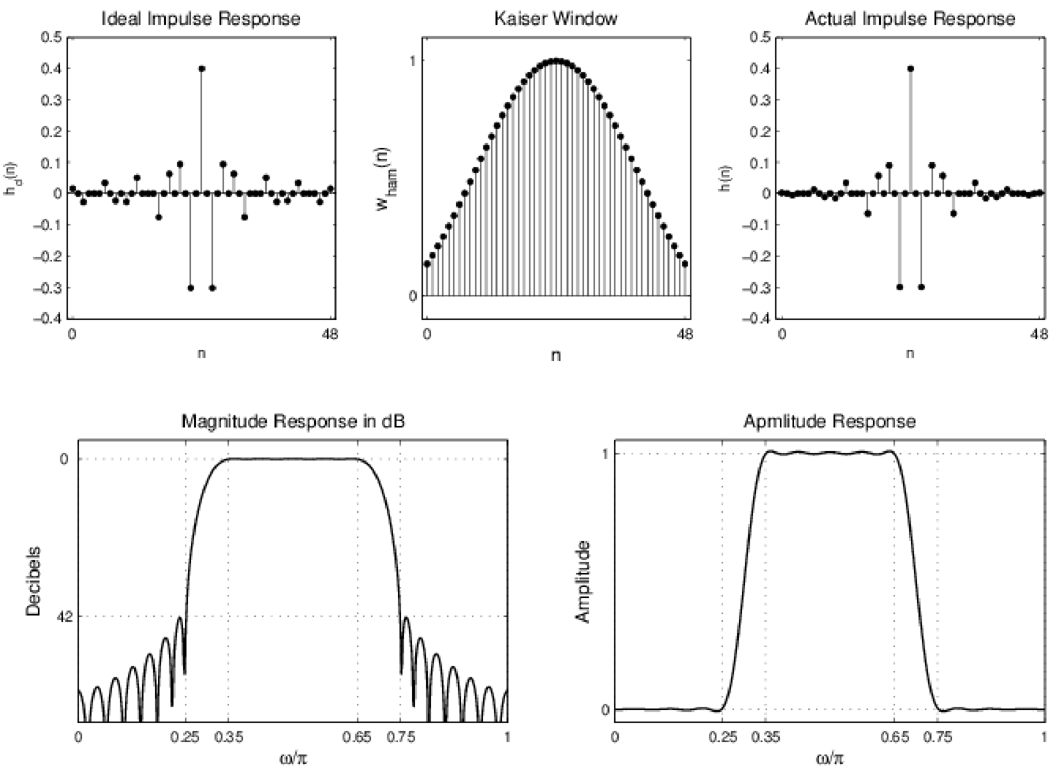


Figure 7.7: Filter design plots in Problem 7.14

P7.15

Design the staircase filter of Example 7.26 using the Kaiser window approach. The specifications are

$$\text{Band-1: } 0 \le \omega \le 0.3\pi, \text{ Ideal gain} = 1, \ \delta_1 = 0.01$$

$$\text{Band-2: } 0.4\pi \le \omega \le 0.7\pi, \text{ Ideal gain} = 0.5, \ \delta_2 = 0.005$$

$$\text{Band-3: } 0.8\pi \le \omega \le \pi, \text{ Ideal gain} = 0, \ \delta_3 = 0.001$$

Compare the filter length of this design with that of Example 7.26. Provide a plot of the magnitude response in dB. Do not use the **fir1** function.

Solutions

```
% P7.15
clear;clc; close all;
%% Specifications:
w1 = 0.0*pi; % lower Band-1 edge
w2 = 0.3*pi; % upper Band-1 edge
w3 = 0.4*pi; % lower Band-2 edge
w4 = 0.7*pi; % upper Band-2 edge
w5 = 0.8*pi; % lower Band-3 edge
w6 = 1.0*pi; % upper Band-3 edge
delta1 = 0.01; % Band-1 ripple
```

```
delta2 = 0.005; % Band-2 ripple
delta3 = 0.001; % Band-3 ripple
%
% Determine Kaiser Window Parameters
delta = min([delta1,delta2,delta3]); tr_width = min([w3-
w2,w5-w4]);
[Rp,As] = delta2db(delta1,delta3);
M = ceil((As-7.95)/(2.285*tr_width)+1)+1; M =
2*floor(M/2)+1, % choose odd M
n = [0:1:M-1]; beta = 0.1102*(As-8.7); w_kai =
(kaiser(M,beta))'; % Kaiser Window
% Determine Ideal Impulse Response
wc1 = (w2+w3)/2; wc2 = (w4+w5)/2;
hd = ideal_lp(wc1,M)+0.5*(ideal_lp(wc2,M)-
ideal_lp(wc1,M));
% Determine the Window Design Impulse Response and
Frequency Response
h = hd .* w_kai; [db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db(ceil(w5/delta_w)+1:501))), % Actual
Attn
[Hr,w,P,L] = ampl_res(h);
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,7,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.15');
subplot('position',[0.08,0.6,0.25,0.35]);
Hs_1= stem(n,hd,'filled'); set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)');
subplot('position',[0.41,0.6,0.25,0.35]);
Hs_2 = stem(n,w_kai,'filled'); set(Hs_2,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Kaiser Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8);
subplot('position',[0.74,0.6,0.25,0.35]);
Hs_3 = stem(n,h,'filled'); set(Hs_3,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8);
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
```

```
ylabel('h(n)');
subplot('position',[0.09,0.1,0.4,0.35]);
plot(w/pi,db,'linewidth',1); title('Magnitude Response in
dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels');
set(gca,'XTick',[0;0.3;0.4;0.7;0.8;1]);
set(gca,'YTick',[-Asd;0]); set(gca,'YTickLabel',{' 65';'
0 '});grid;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,Hr,'linewidth',1); title('Apmlitude Response');
axis([0,1,-0.05,1.05]); xlabel('\omega/\pi');
ylabel('Amplitude');
set(gca,'XTick',[0;0.3;0.4;0.7;0.8;1]);
set(gca,'YTick',[0;0.5;1]); grid;
print -deps2 ../EPSFILES/P0715

M =
    75
Asd =
    65
*** Type-1 Linear-Phase Filter ***
```

The window-designed filter has the length of 75 while the one designed in Example 7.26 has the length of 49. The filter response plots are shown in Figure 7.8.

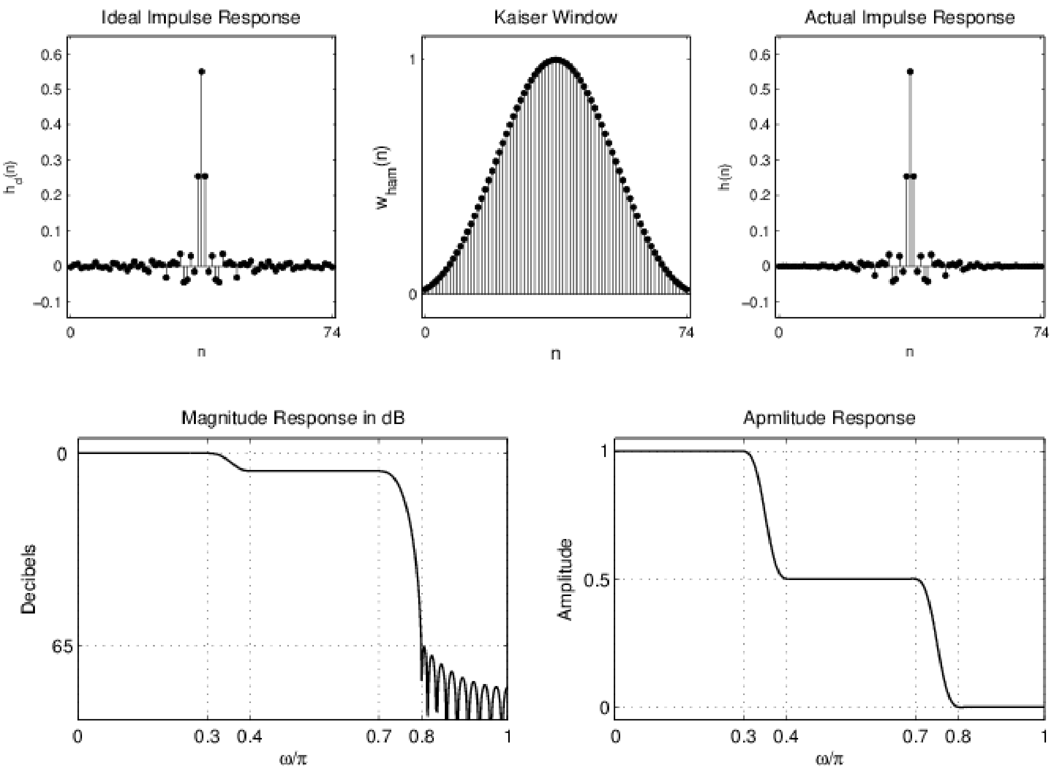


Figure 7.8: Filter design plots in Problem 7.15

P7.16

Design a bandpass filter using a fixed window design technique that has the minimum length and that satisfies the following specifications:

lower stopband edge = 0.3π
upper stopband edge = 0.6π $A_s = 40$ dB
lower passband edge = 0.4π
upper passband edge = 0.5π $R_p = 0.5$ dB.

Provide a plot of the log-magnitude response in dB and **stem** plot of the impulse response.

Solutions

```
% P7.16
clear;clc; close all;
%% Specifications:
ws1 = 0.3*pi; % lower stopband edge
wp1 = 0.4*pi; % lower passband edge
wp2 = 0.5*pi; % upper passband edge
ws2 = 0.6*pi; % upper stopband edge
Rp = 0.5; % passband ripple
As = 40; % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
% Determine Kaiser Window Parameters
tr_width = abs(min((wp1-ws1),(ws2-wp2)));
M = ceil((As-7.95)/(2.285*tr_width)+1)+1; M =
2*floor(M/2)+3, % choose odd M

n = [0:1:M-1]; beta = 0.1102*(As-8.7); w_kai =
(kaiser(M,beta))'; % Kaiser Window
% Determine Ideal Impulse Response
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(wc2,M)-ideal_lp(wc1,M);
% Determine the Window Design Impulse Response and
Frequency Response
h = hd .* w_kai; [db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
```

```
Rpd = -min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w))), %
Actual passband ripple

Asd = floor(-max(db(1:floor(ws1/delta_w)))), % Actual
Attn

[Hr,w,P,L] = ampl_res(h);
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,7,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.16');
subplot('position',[0.08,0.6,0.25,0.35]);
Hs_1= stem(n,hd,'filled'); set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)');
subplot('position',[0.41,0.6,0.25,0.35]);
Hs_2 = stem(n,w_kai,'filled'); set(Hs_2,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Kaiser Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8);
subplot('position',[0.74,0.6,0.25,0.35]);
Hs_3 = stem(n,h,'filled'); set(Hs_3,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8);
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)');
subplot('position',[0.09,0.1,0.4,0.35]);
plot(w/pi,db,'linewidth',1); title('Magnitude Response in
dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels');
set(gca,'XTick',[0;0.3;0.4;0.5;0.6;1])
set(gca,'YTick',[-Asd;0]); set(gca,'YTickLabel',{' 40';'
0 '});grid;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,Hr,'linewidth',1); title('Apmlitude Response');
axis([0,1,-0.05,1.05]); xlabel('\omega/\pi');
ylabel('Amplitude');
set(gca,'XTick',[0;0.3;0.4;0.5;0.6;1]);
set(gca,'YTick',[0;1]); grid;
```

```
print -deps2 ../EPSFILES/P0716

M =
    49
Rpd =
    0.1872
Asd =
    40
*** Type-1 Linear-Phase Filter ***
```

The filter response plots are shown in Figure 7.9.

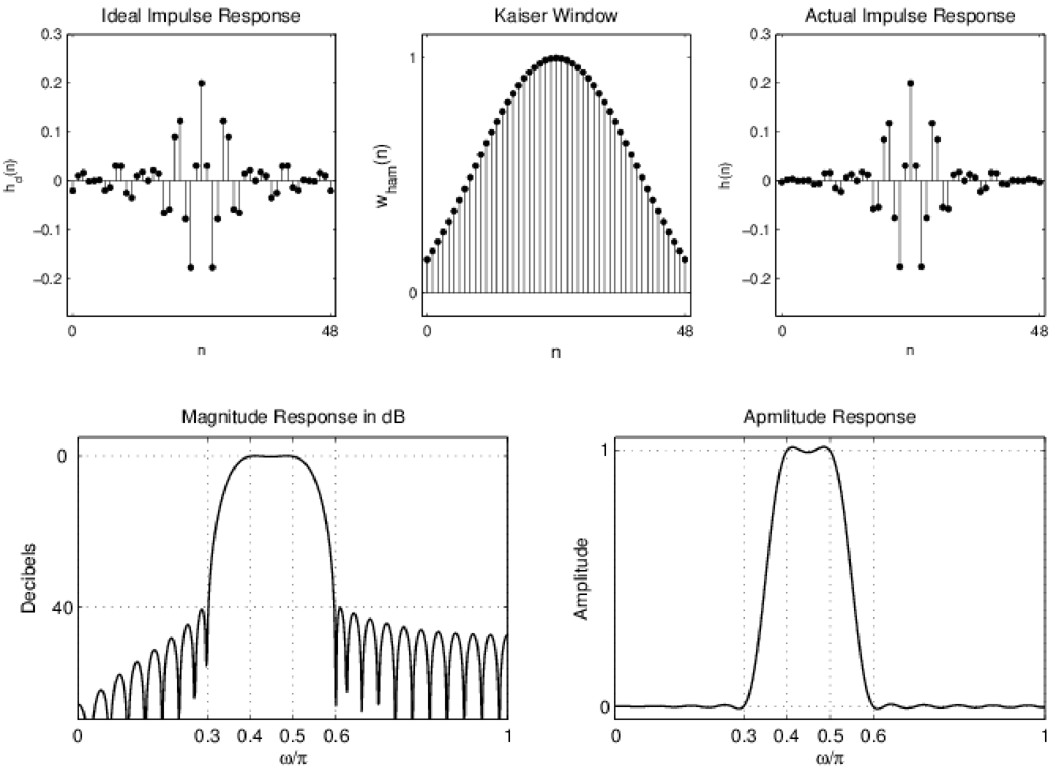


Figure 7.9: Filter design plots in Problem 7.16

P7.17

Repeat Problem P7.9 using the **fir1** function.

Solutions

```
% P7.17
clear;clc; close all;
%% Specifications:
ws1 = 0.2*pi; % lower stopband edge
wp1 = 0.35*pi; % lower passband edge
wp2 = 0.55*pi; % upper passband edge
```

```
ws2 = 0.75*pi; % upper stopband edge
Rp = 0.25; % passband ripple
As = 40; % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = min((wp1-ws1),(ws2-wp2));
M = ceil(6.2*pi/tr_width); M = 2*floor(M/2)+1, % choose
odd M

n = 0:M-1; w_han = (hann(M))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2; hd =
ideal_lp(wc2,M)-ideal_lp(wc1,M);
h = fir1(M-1,[wc1,wc2]/pi,'bandpass',w_han);
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Rpd = -min(db((wp1/delta_w)+1:(wp2/delta_w)+1)), % Actual
passband ripple

Asd = floor(-max(db(1:(ws1/delta_w)+1))), % Actual Attn

%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.17');
subplot(2,2,1); Hs_1= stem(n,hd,'filled');
set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)')
subplot(2,2,2); Hs_1 = stem(n,w_han,'filled');
set(Hs_1,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{han}(n)');
title('Hann Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8)
subplot(2,2,3); Hs_1 = stem(n,h,'filled');
```

```
set(Hs_1,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)')
subplot(2,2,4); plot(w/pi,db,'linewidth',1);
title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels')
set(gca,'XTick',[0;0.2;0.35;0.55;0.75;1])
set(gca,'XTickLabel',{' 0 ';'0.2 ';'0.35';'0.55';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[-40;0]); set(gca,'YTickLabel',{' 40';' 0
'});grid
print -deps2 ../EPSFILES/P0717

M =
    43
Rpd =
    0.1030
Asd =
    44
```

The filter response plots are shown in Figure 7.10.

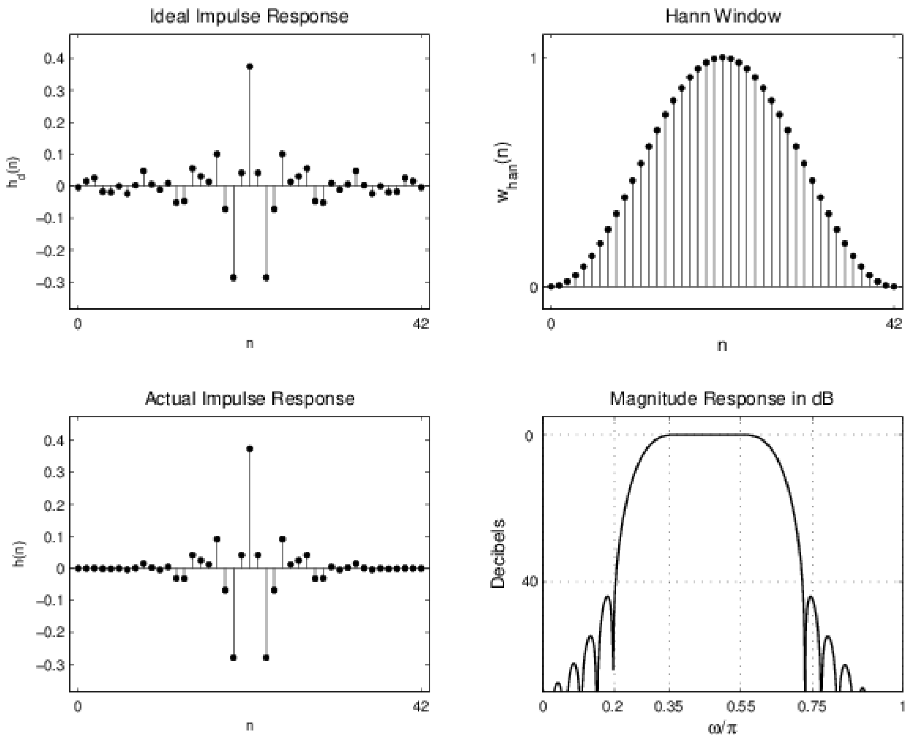


Figure 7.10: Filter design plots in Problem 7.17

P7.18

Repeat Problem P7.10 using the **fir1** function.

Solutions

```
% P7.18
clear;clc; close all;
%% Specifications:
wp1 = 0.3*pi; % lower passband edge
ws1 = 0.4*pi; % lower stopband edge
ws2 = 0.6*pi; % upper stopband edge
wp2 = 0.7*pi; % upper passband edge
Rp = 0.2; % passband ripple
As = 50; % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = abs(min((wp1-ws1),(ws2-wp2)));
M = ceil(6.6*pi/tr_width); M = 2*floor(M/2)+1, % choose
odd M

n = 0:M-1; w_ham = (hamming(M))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2; hd =
ideal_lp(pi,M)+ideal_lp(wc1,M)-ideal_lp(wc2,M);
h = fir1(M-1,[wc1,wc2]/pi,'stop',w_ham);
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Asd = floor(-
max(db(ceil(ws1/delta_w)+1:floor(ws2/delta_w)+1))), %
Actual Attn

Rpd = -min(db(1:floor(wp1/delta_w)+1)), % Actual passband
ripple

%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
```

```
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.18');
subplot(2,2,1); Hs_1= stem(n,hd,'filled');
set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)')
subplot(2,2,2); Hs_1 = stem(n,w_ham,'filled');
set(Hs_1,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Hamming Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8)
subplot(2,2,3); Hs_1 = stem(n,h,'filled');
set(Hs_1,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)')
subplot(2,2,4); plot(w/pi,db,'linewidth',1);
title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels')
set(gca,'XTick',[0;0.3;0.4;0.6;0.7;1])
set(gca,'XTickLabel',{' 0 ';'0.3';'0.4';'0.6';'0.7';' 1
'},'fontsize',8)
set(gca,'YTick',[-50;0]); set(gca,'YTickLabel',{' 50';' 0
'});grid
print -deps2 ../EPSFILES/P0718

M =
    67
Asd =
    50
Rpd =

    0.0435
```

The filter response plots are shown in Figure 7.11.

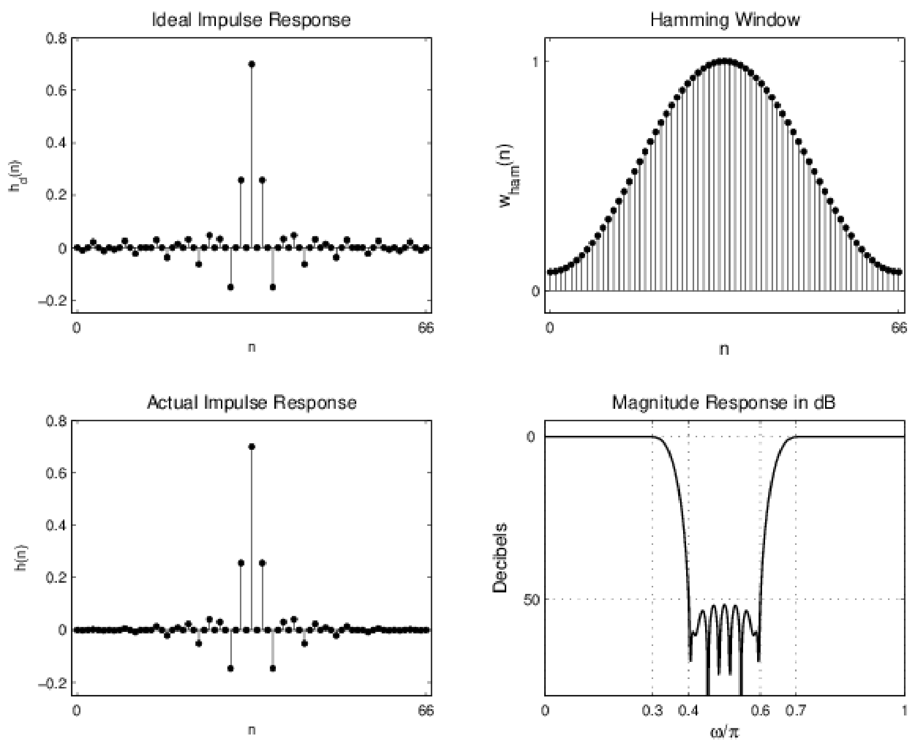


Figure 7.11: Filter design plots in Problem 7.18

P7.19

Repeat Problem P7.11 using the **fir1** function.

Solutions

```
% P7.19
clear;clc; close all;
%% Specifications:
ws1 = 0.3*pi; % lower stopband edge
wp1 = 0.4*pi; % lower passband edge
wp2 = 0.5*pi; % upper passband edge
ws2 = 0.6*pi; % upper stopband edge
Rp = 0.5; % passband ripple
As = 50; % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
```

```
%
tr_width = abs(min((wp1-ws1),(ws2-wp2)));
M = ceil(6.6*pi/tr_width); M = 2*floor(M/2)+1, % choose
odd M

n = 0:M-1; w_ham = (hamming(M))';
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2; hd =
ideal_lp(wc2,M)-ideal_lp(wc1,M);
h = fir1(M-1,[wc1,wc2]/pi,'bandpass',w_ham);
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Rpd = -
min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w)+1)), %
Actual passband ripple

Asd = floor(-max(db(1:(ws1/delta_w)+1))), % Actual Attn

%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.19');
subplot(2,2,1); Hs_1= stem(n,hd,'filled');
set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)')
subplot(2,2,2); Hs_1 = stem(n,w_ham,'filled');
set(Hs_1,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Hamming Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8)
subplot(2,2,3); Hs_1 = stem(n,h,'filled');
set(Hs_1,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)')
subplot(2,2,4); plot(w/pi,db,'linewidth',1);
title('Magnitude Response in dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels')
set(gca,'XTick',[0;0.3;0.4;0.5;0.6;1])
```

```
set(gca,'XTickLabel',{' 0 ';'0.3';'0.4';'0.5';'0.6';' 1
'},'fontsize',8)
set(gca,'YTick',[-50;0]); set(gca,'YTickLabel',{' 50';' 0
'});grid
print -deps2 ../EPSFILES/P0719

M =
    67
Rpd =
    0.0488
Asd =
    51
```

The filter response plots are shown in Figure 7.12.

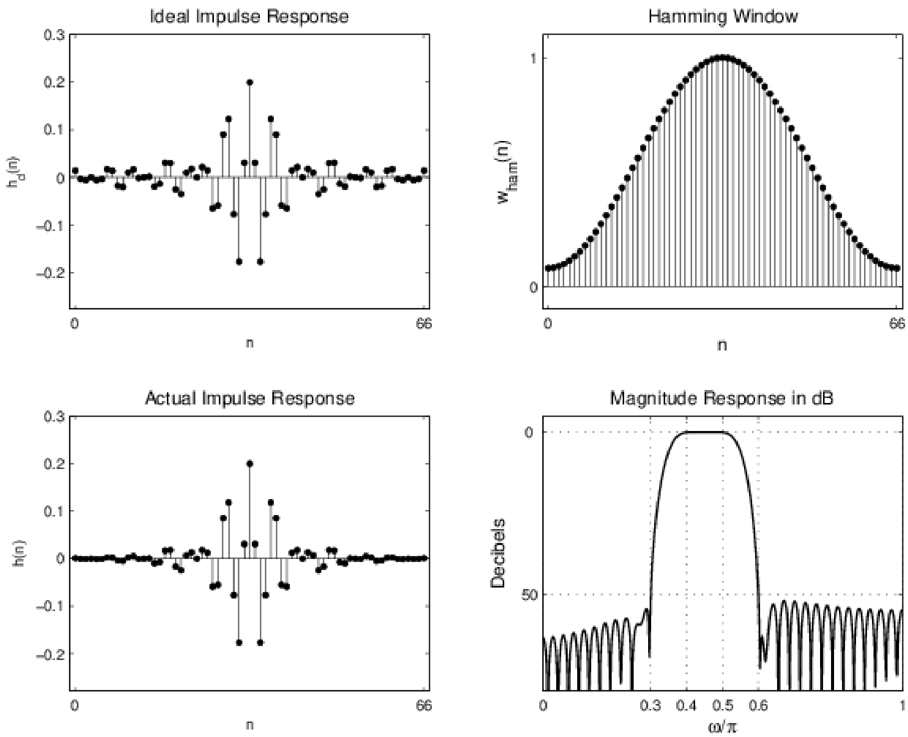


Figure 7.12: Filter design plots in Problem 7.19

P7.20

Repeat Problem P7.12 using the **fir1** function.

Solutions

```
% P7.20
clear; clc; close all;
%% Specifications:
ws = 0.4*pi; % stopband edge
```

```
wp = 0.6*pi; % passband edge
Rp = 0.001; % passband ripple
As = 50; % stopband attenuation
%
% Select the min(delta1,delta2) since delta1=delta2 in
windodow design
[delta1,delta2] = db2delta(Rp,As);
if (delta1 < delta2)
delta2 = delta1; disp('Delta1 is smaller than delta2')
[Rp,As] = delta2db(delta1,delta2)
end
%
tr_width = abs(wp-ws);
M = ceil(11*pi/tr_width); M = 2*floor(M/2)+1, % choose
odd M

n = 0:M-1; w_blk = (blackman(M))';
wc = (ws+wp)/2; hd = ideal_lp(pi,M)-ideal_lp(wc,M);
h = fir1(M-1,wc/pi,'high',w_blk);
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Rpd = -min(db(ceil(wp/delta_w)+1:floor(pi/delta_w)+1)), %
Actual passband ripple

Asd = floor(-max(db(1:(ws/delta_w)+1))), % Actual Attn

%
%% Zoomed Filter Response Plot
Hf_1 = figure('Units','inches','position',[1,1,5,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.20');
plot(w(301:501)/pi,db(301:501),'linewidth',1);
title('Zoomed Magnitude Response in dB');
axis([0.6,1,-0.005,0.001]); xlabel('\omega/\pi');
ylabel('Decibels')
set(gca,'XTick',[0.6;1])
set(gca,'XTickLabel',{'0.6';' 1 '},'fontsize',8)
set(gca,'YTick',[-0.004;0]); set(gca,'YTickLabel',{'-
0.004';' 0 '});grid
print -deps2 ../EPSFILES/P0720

Delta1 is smaller than delta2
Rp =
   1.0000e-03
As =
```

```
   84.7974
M =
    55
Rpd =
    0.0039
Asd =
    71
```

The zoomed magnitude response plot is shown in Figure 7.13.

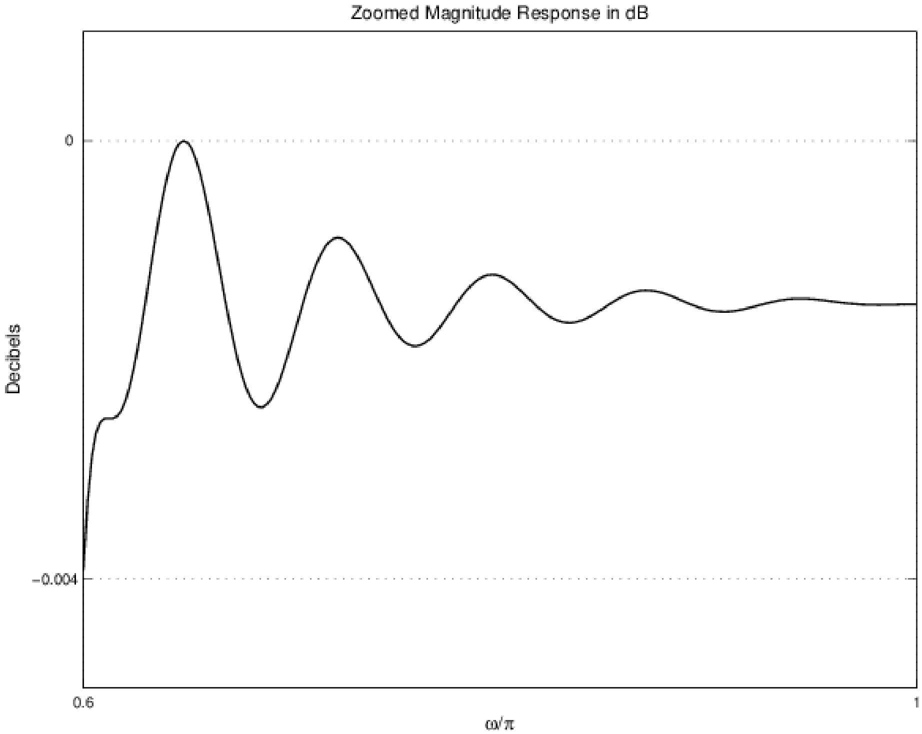


Figure 7.13: Filter design plots in Problem 7.20

P7.21

Repeat Problem P7.13 using the **fir1** function.

Solutions

```
% P7.21
clear;clc; close all;
%% Specifications:
wp1 = 0.25*pi; % lower passband edge
ws1 = 0.35*pi; % lower stopband edge
ws2 = 0.65*pi; % upper stopband edge
wp2 = 0.75*pi; % upper passband edge
delta1 = 0.025; % passband ripple
delta2 = 0.005; % stopband ripple
```

```
%
% Convert to decibels
[Rp,As] = delta2db(delta1,delta2)

%
tr_width = abs(min((wp1-ws1),(ws2-wp2))); M = ceil((As-
7.95)/(2.285*tr_width)+1)+1;
M = 2*floor(M/2)+1, % choose odd M

n = [0:1:M-1]; beta = 0.1102*(As-8.7); w_kai =
(kaiser(M,beta))'; % Kaiser Window
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(pi,M)+ideal_lp(wc1,M)-ideal_lp(wc2,M); %
Ideal HP Filter
h = fir1(M-1,[wc1,wc2]/pi,'stop',w_kai);
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Asd = floor(-
max(db(ceil(ws1/delta_w)+1:floor(ws2/delta_w)+1))), %
Actual Attn

Rpd = -min(db(1:floor(wp1/delta_w)+1)), % Actual passband
ripple

[Hr,w,P,L] = ampl_res(h);
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,7,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.21');
subplot('position',[0.08,0.6,0.25,0.35]);
Hs_1= stem(n,hd,'filled'); set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)');
subplot('position',[0.41,0.6,0.25,0.35]);
Hs_2 = stem(n,w_kai,'filled'); set(Hs_2,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Kaiser Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8);
subplot('position',[0.74,0.6,0.25,0.35]);
Hs_3 = stem(n,h,'filled'); set(Hs_3,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
```

```
1],'fontsize',8);
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)');
subplot('position',[0.09,0.1,0.4,0.35]);
plot(w/pi,db,'linewidth',1); title('Magnitude Response in
dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
ylabel('Decibels');
set(gca,'XTick',[0;0.25;0.35;0.65;0.75;1])
set(gca,'XTickLabel',{' 0 ';'0.25';'0.35';'0.65';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[-Asd;0]); set(gca,'YTickLabel',{' 49';'
0 '});grid;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,Hr,'linewidth',1); title('Apmlitude Response');
axis([0,1,-0.05,1.05]); xlabel('\omega/\pi');
ylabel('Amplitude');
set(gca,'XTick',[0;0.25;0.35;0.65;0.75;1]);
set(gca,'XTickLabel',{' 0 ';'0.25';'0.35';'0.65';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[0;1]); grid;
print -deps2 ../EPSFILES/P0721

Rp =
    0.4344
As =
   46.2351
M =
    57
Asd =
    49
Rpd =
    0.0492
*** Type-1 Linear-Phase Filter ***
```

The filter response plots are shown in Figure 7.14.

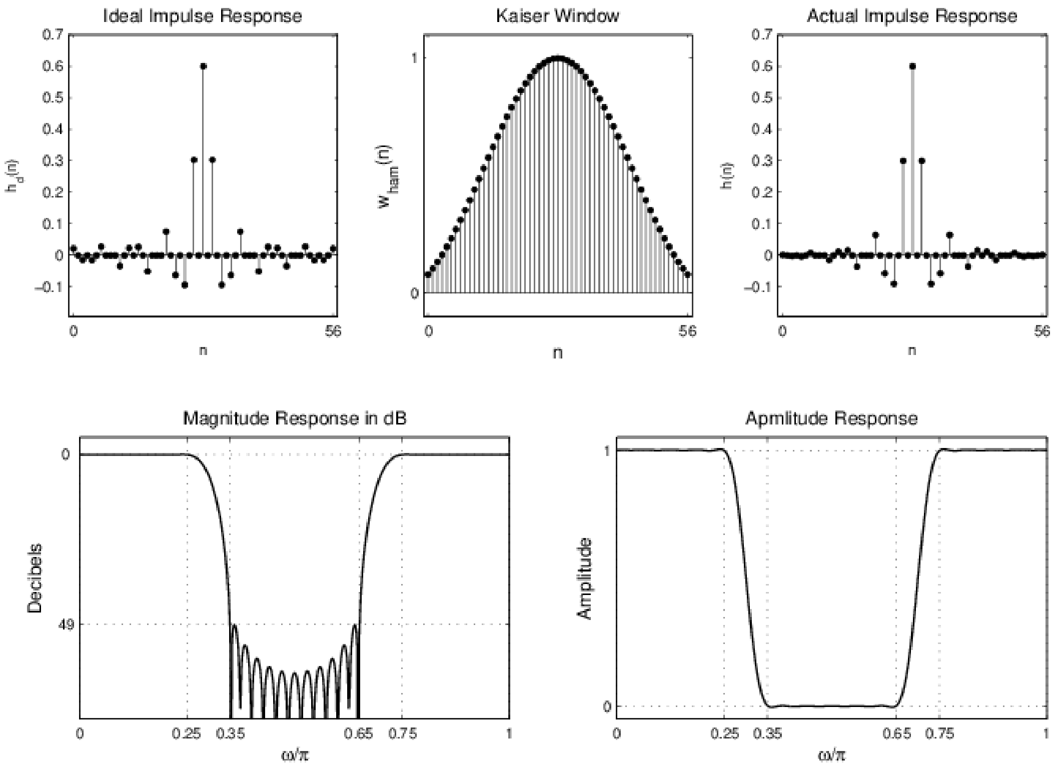


Figure 7.14: Filter design plots in Problem 7.21

P7.22

Repeat Problem P7.14 using the **fir1** function.

Solutions

```
% P7.22
clear;clc; close all;
%% Specifications:
ws1 = 0.25*pi; % lower stopband edge
wp1 = 0.35*pi; % lower passband edge
wp2 = 0.65*pi; % upper passband edge
ws2 = 0.75*pi; % upper stopband edge
delta1 = 0.05; % passband ripple
delta2 = 0.01; % stopband ripple
%
% Convert to decibels
[Rp,As] = delta2db(delta1,delta2)

%
tr_width = abs(min((wp1-ws1),(ws2-wp2)));
M = ceil((As-7.95)/(2.285*tr_width)+1)+1; M =
```

```
2*floor(M/2)+1, % choose odd M

n = [0:1:M-1]; beta = 0.1102*(As-8.7); w_kai =
(kaiser(M,beta))'; % Kaiser Window
wc1 = (ws1+wp1)/2; wc2 = (ws2+wp2)/2;
hd = ideal_lp(wc2,M)-ideal_lp(wc1,M);
% Determine the Window Design Impulse Response and
Frequency Response
h = fir1(M-1,[wc1,wc2]/pi,'bandpass',w_kai);
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Rpd = -min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w))), %
Actual passband ripple

Asd = floor(-max(db(1:floor(ws1/delta_w)+1))), % Actual
Attn

[Hr,w,P,L] = ampl_res(h);
%
%% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,7,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.22');
subplot('position',[0.08,0.6,0.25,0.35]);
Hs_1= stem(n,hd,'filled'); set(Hs_1,'markersize',3);
title('Ideal Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8)
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h_d(n)');
subplot('position',[0.41,0.6,0.25,0.35]);
Hs_2 = stem(n,w_kai,'filled'); set(Hs_2,'markersize',3);
axis([-1,M,-0.1,1.1]); xlabel('n'); ylabel('w_{ham}(n)');
title('Kaiser Window');
set(gca,'XTick',[0;M-1],'fontsize',8);
set(gca,'YTick',[0;1],'fontsize',8);
subplot('position',[0.74,0.6,0.25,0.35]);
Hs_3 = stem(n,h,'filled'); set(Hs_3,'markersize',3);
title('Actual Impulse Response'); set(gca,'XTick',[0;M-
1],'fontsize',8);
axis([-1,M,min(hd)-0.1,max(hd)+0.1]); xlabel('n');
ylabel('h(n)');
subplot('position',[0.09,0.1,0.4,0.35]);
plot(w/pi,db,'linewidth',1); title('Magnitude Response in
dB');
axis([0,1,-As-30,5]); xlabel('\omega/\pi');
```

```
ylabel('Decibels');
set(gca,'XTick',[0;0.25;0.35;0.65;0.75;1])
set(gca,'XTickLabel',{' 0 ';'0.25';'0.35';'0.65';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[-Asd;0]); set(gca,'YTickLabel',{' 42';'
0 '});grid;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,Hr,'linewidth',1); title('Apmlitude Response');
axis([0,1,-0.05,1.05]); xlabel('\omega/\pi');
ylabel('Amplitude');
set(gca,'XTick',[0;0.25;0.35;0.65;0.75;1]);
set(gca,'XTickLabel',{' 0 ';'0.25';'0.35';'0.65';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[0;1]); grid;
print -deps2 ../EPSFILES/P0722

Rp =
    0.8693
As =
   40.4238
M =
    49
Rpd =
    0.1033
Asd =
    42
*** Type-1 Linear-Phase Filter ***
```

The filter response plots are shown in Figure 7.15.

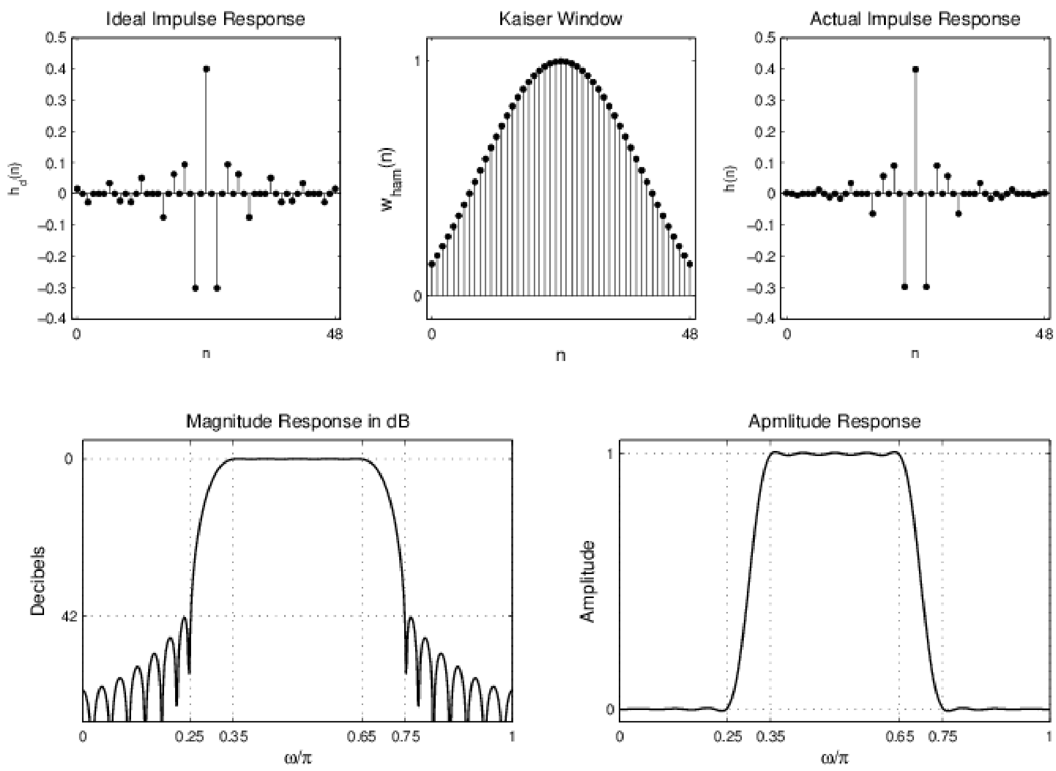


Figure 7.15: Filter design plots in Problem 7.22

P7.23

Consider an ideal lowpass filter with the cutoff frequency $\omega_c = 0.3\pi$. We want to approximate this filter using a frequency sampling design in which we choose 40 samples.

1. Choose the sample at ω_c equal to 0.5, and use the naive design method to compute $h(n)$. Determine the minimum stopband attenuation.
2. Now vary the sample at ω_c, and determine the optimum value to obtain the largest minimum stopband attenuation.
3. Plot the magnitude responses in dB of the preceding two designs in one plot, and comment on the results.

P7.24

Design the bandstop filter of Problem P7.10 using the frequency sampling method. Choose the order of the filter appropriately so that there are two samples in the transition band. Use optimum values for these samples. Compare your results with those obtained using the **fir2** function.

Solutions

```
% P7.24
clear;clc; close all;
```

```
% Specifications:
wp1 = 0.3*pi; % lower passband edge
ws1 = 0.4*pi; % lower stopband edge
ws2 = 0.6*pi; % upper stopband edge
wp2 = 0.7*pi; % upper passband edge
Rp = 0.2; % passband ripple
As = 50; % stopband attenuation
Hf_1 = figure('Units','inches','position',[1,1,7,5],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.24');
%% Frequency Sampling Design
% Choose M = 65 to get two samples in the transition band
M = 65; alpha = (M-1)/2; k = 0:M-1; wk = (2*pi/M)*k;
% Samples of the Frequency Response
T1 = 0.58753; T2 = 0.10508;
Hrs = [ones(1,11),T1,T2,zeros(1,8),T2,T1,ones(1,20),...
T1,T2,zeros(1,8),T2,T1,ones(1,10)];
% Ideal Amplitude Response for Plotting
fc1 = (wp1+ws1)/(2*pi); fc2 = (ws2+wp2)/(2*pi);
Hdr = [1,1,0,0,1,1]; fdr = [0,fc1,fc1,fc2,fc2,1];
% Compute the Impulse Response
k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
H = Hrs.*exp(j*angH); h = real(ifft(H,M));
% Compute Frequency Responses
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
% Actual Attn
Asd = floor(-
max(db(ceil(ws1/delta_w)+1:floor(ws2/delta_w)+1)));
% Filter Response Plots
subplot('position',[0.09,0.6,0.4,0.35]);
Hp_1 = plot(wk(1:(M+1)/2)/pi,Hrs(1:(M+1)/2),'mo'); hold
on;
set(Hp_1,'markersize',3);
plot(fdr,Hdr,'g','linewidth',2); axis([0,1,0,1.1]);
plot(w/pi,mag,'r','linewidth',1.5);
xlabel('\omega/\pi'); ylabel('Magnitude');
title('Frequency Sampling Design: Magnitude Plot');
set(gca,'xtick',[0,wp1,ws1,ws2,wp2,pi]/pi,'xgrid','on');
set(gca,'ytick',[0,1]); hold off;
subplot('position',[0.59,0.6,0.4,0.35]);
plot(w/pi,db,'g','linewidth',2); axis([0,1,-80,10]);
xlabel('\omega/\pi'); ylabel('Decibels');
title('Frequency Sampling Design: Log-Magnitude Plot');
```

```
set(gca,'xtick',[0,wp1,ws1,ws2,wp2,pi]/pi,'xgrid','on');
set(gca,'ytick',[-Asd,0],'ygrid','on');
%% Design Using the FIR2 Function
fc1 = (wp1+ws1)/(2*pi); fc2 = (ws2+wp2)/(2*pi);
Hdr = [1,1,0,0,1,1]; fdr = [0,fc1,fc1,fc2,fc2,1];
h = fir2(82,fdr,Hdr);
% Compute Frequency Responses
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
% Actual Attn
Asd = floor(-
max(db(ceil(ws1/delta_w)+1:floor(ws2/delta_w)+1)));
% Filter Response Plots
subplot('position',[0.09,0.1,0.4,0.35]);
Hp_2 = plot(wk(1:(M+1)/2)/pi,Hrs(1:(M+1)/2),'mo'); hold
on;set(Hp_2, 'markersize',3);
plot(fdr,Hdr,'g','linewidth',2); axis([0,1,0,1.1]);
plot(w/pi,mag,'r','linewidth',1.5);
xlabel('\omega/\pi'); ylabel('Magnitude');
title('FIR2 Function Design: Magnitude Plot');
set(gca,'xtick',[0,wp1,ws1,ws2,wp2,pi]/pi,'xgrid','on');
set(gca,'ytick',[0,1]); hold off;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,db,'g','linewidth',2); axis([0,1,-80,10]);
xlabel('\omega/\pi'); ylabel('Decibels');
title('FIR2 Function Design: Log-Magnitude Plot');
set(gca,'xtick',[0,wp1,ws1,ws2,wp2,pi]/pi,'xgrid','on');
set(gca,'ytick',[-Asd,0],'ygrid','on');
print -deps2 ../EPSFILES/P0724
```

The filter response plots are shown in Figure 7.16.

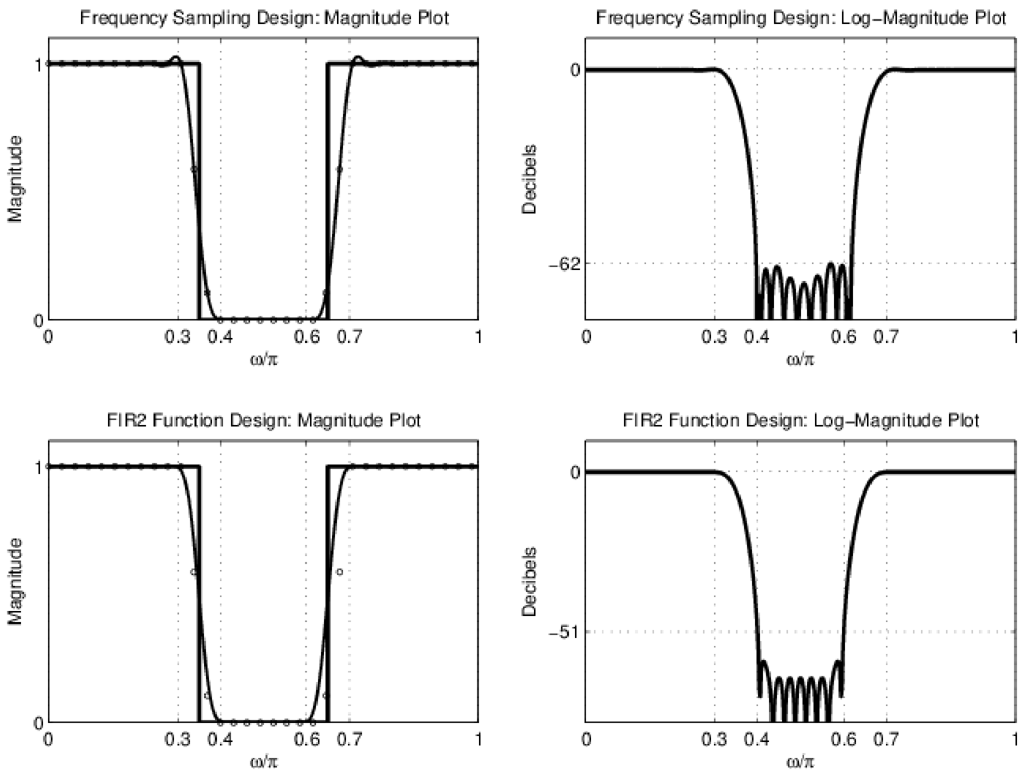


Figure 7.16: Filter design plots in Problem 7.24

P7.25

Design the bandpass filter of Problem P7.11 using the frequency sampling method. Choose the order of the filter appropriately so that there are two samples in the transition band. Use optimum values for these samples. Compare your results with those obtained using the **fir2** function.

Solutions

```
% P7.25
clear;clc; close all;
% Specifications:
ws1 = 0.3*pi; % lower stopband edge
wp1 = 0.4*pi; % lower passband edge
wp2 = 0.5*pi; % upper passband edge
ws2 = 0.6*pi; % upper stopband edge
Rp = 0.5; % passband ripple
As = 50; % stopband attenuation
Hf_1 = figure('Units','inches','position',[1,1,7,5],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.25');
%% Frequency Sampling Design
```

```
% Choose M = 60 to get two samples in the transition band
M = 60; alpha = (M-1)/2; k = 0:M-1; wk = (2*pi/M)*k;
% Samples of the Frequency Response
T0 = 1; T1 = 0.58753; T2 = 0.10508; T3 = 0;
TL = [T3,T2,T1,T0]; TR = fliplr(TL);
Hrs = [zeros(1,9),TL,ones(1,2),TR,zeros(1,23),...
TL,ones(1,2),TR,zeros(1,8)];
% Ideal Amplitude Response for Plotting
fc1 = (wp1+ws1)/(2*pi); fc2 = (ws2+wp2)/(2*pi);
Hdr = [0,0,1,1,0,0]; fdr = [0,fc1,fc1,fc2,fc2,1];
% Compute the Impulse Response
k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
H = Hrs.*exp(j*angH); h = real(ifft(H,M));
% Compute Frequency Responses
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
% Actual Attn
Asd = floor(-max(db(1:floor(ws1/delta_w))));
% Filter Response Plots
subplot('position',[0.09,0.6,0.4,0.35]);
Hp_1 =
plot(wk(1:floor((M+1)/2))/pi,Hrs(1:floor((M+1)/2)),'mo');
set(Hp_1, 'markersize',3);
hold on; plot(fdr,Hdr,'g','linewidth',2);
axis([0,1,0,1.1]);
plot(w/pi,mag,'r','linewidth',1.5);
xlabel('\omega/\pi'); ylabel('Magnitude');
title('Frequency Sampling Design: Magnitude Plot');
set(gca,'xtick',[0,ws1,wp1,wp2,ws2,pi]/pi,'xgrid','on','y
grid','on');
set(gca,'ytick',[0,1]); hold off;
subplot('position',[0.59,0.6,0.4,0.35]);
plot(w/pi,db,'g','linewidth',2); axis([0,1,-80,10]);
xlabel('\omega/\pi'); ylabel('Decibels');
title('Frequency Sampling Design: Log-Magnitude Plot');
set(gca,'xtick',[0,ws1,wp1,wp2,ws2,pi]/pi,'xgrid','on');
set(gca,'ytick',[-Asd,0],'ygrid','on');
%% Design Using the FIR2 Function
fc1 = (wp1+ws1)/(2*pi); fc2 = (ws2+wp2)/(2*pi);
Hdr = [0,0,1,1,0,0]; fdr = [0,fc1,fc1,fc2,fc2,1];
h = fir2(78,fdr,Hdr);
% Compute Frequency Responses
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
% Actual Attn
```

```
Asd = floor(-max(db(1:floor(ws1/delta_w))));
% Filter Response Plots
subplot('position',[0.09,0.1,0.4,0.35]);
Hp_2 =
plot(wk(1:floor((M+1)/2))/pi,Hrs(1:floor((M+1)/2)),'mo');
set(Hp_2,'markersize',3);
hold on; plot(fdr,Hdr,'g','linewidth',2);
axis([0,1,0,1.1]);
plot(w/pi,mag,'r','linewidth',1.5);
xlabel('\omega/\pi'); ylabel('Magnitude');
title('FIR2 Function Design: Magnitude Plot');
set(gca,'xtick',[0,ws1,wp1,wp2,ws2,pi]/pi,'xgrid','on','y
grid','on');
set(gca,'ytick',[0,1]); hold off;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,db,'g','linewidth',2); axis([0,1,-80,10]);
xlabel('\omega/\pi'); ylabel('Decibels');
title('FIR2 Function Design: Log-Magnitude Plot');
set(gca,'xtick',[0,ws1,wp1,wp2,ws2,pi]/pi,'xgrid','on');
set(gca,'ytick',[-Asd,0],'ygrid','on');
print -deps2 ../EPSFILES/P0725
```

The filter response plots are shown in Figure 7.17.

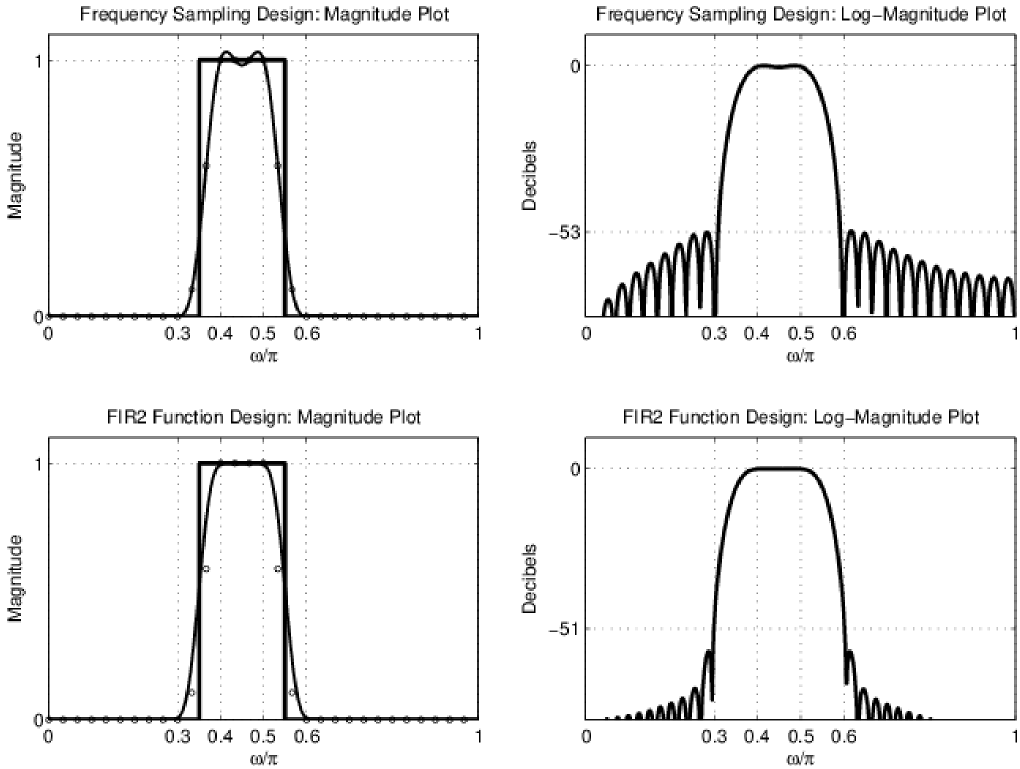


Figure 7.17: Filter design plots in Problem 7.25

P7.26

Design the highpass filter of Problem P7.12 using the frequency sampling method. Choose the order of the filter appropriately so that there are two samples in the transition band. Use optimum values. Compare your results with those obtained using the **fir2** function.

Solutions

```
% P7.26
clear;clc; close all;
% Specifications:
ws = 0.3*pi; % Stopband edge
wp = 0.4*pi; % Passband edge
Rp = 0.004; % passband ripple
As = 50; % stopband attenuation
Hf_1 = figure('Units','inches','position',[1,1,7,5],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.26');
%% Frequency Sampling Design
% Choose M = 65 to get two samples in the transition band
M = 65; alpha = (M-1)/2; k = 0:M-1; wk = (2*pi/M)*k;
% Samples of the Frequency Response
T1 = 0.58753; T2 = 0.10508;
Hrs = [zeros(1,11),T2,T1,ones(1,40),T1,T2,zeros(1,10)];
% Ideal Amplitude Response for Plotting
fc = (wp+ws)/(2*pi); Hdr = [0,0,1,1]; fdr = [0,fc,fc,1];
% Compute the Impulse Response
k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
H = Hrs.*exp(j*angH); h = real(ifft(H,M));
% Compute Frequency Responses
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
% Actual Attn
Asd = floor(-max(db(1:floor(ws/delta_w))));
% Filter Response Plots
subplot('position',[0.09,0.6,0.4,0.35]);
Hp_1 = plot(wk(1:(M+1)/2)/pi,Hrs(1:(M+1)/2),'mo');
set(Hp_1,'markersize',3);
hold on; plot(fdr,Hdr,'g','linewidth',2);
axis([0,1,0,1.1]);
plot(w/pi,mag,'r','linewidth',1.5);
xlabel('\omega/\pi'); ylabel('Magnitude');
title('Frequency Sampling Design: Magnitude Plot');
```

```
set(gca,'xtick',[0,ws,wp,pi]/pi,'xgrid','on','ygrid','on'
);
set(gca,'ytick',[0,1]); hold off;
subplot('position',[0.59,0.6,0.4,0.35]);
plot(w/pi,db,'g','linewidth',2); axis([0,1,-80,10]);
xlabel('\omega/\pi'); ylabel('Decibels');
title('Frequency Sampling Design: Log-Magnitude Plot');
set(gca,'xtick',[0,ws,wp,pi]/pi,'xgrid','on');
set(gca,'ytick',[-Asd,0],'ygrid','on');
%% Design Using the FIR2 Function
fc = (wp+ws)/(2*pi); Hdr = [0,0,1,1]; fdr = [0,fc,fc,1];
h = fir2(78,fdr,Hdr);
% Compute Frequency Responses
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
% Actual Attn
Asd = floor(-max(db(1:floor(ws/delta_w))));
% Filter Response Plots
subplot('position',[0.09,0.1,0.4,0.35]);
Hp_2 = plot(wk(1:(M+1)/2)/pi,Hrs(1:(M+1)/2),'mo');
set(Hp_2,'markersize',3);
hold on; plot(fdr,Hdr,'g','linewidth',2);
axis([0,1,0,1.1]);
plot(w/pi,mag,'r','linewidth',1.5);
xlabel('\omega/\pi'); ylabel('Magnitude');
title('FIR2 Function Design: Magnitude Plot');
set(gca,'xtick',[0,ws,wp,pi]/pi,'xgrid','on','ygrid','on'
);
set(gca,'ytick',[0,1]); hold off;
subplot('position',[0.59,0.1,0.4,0.35]);
plot(w/pi,db,'g','linewidth',2); axis([0,1,-80,10]);
xlabel('\omega/\pi'); ylabel('Decibels');
title('FIR2 Function Design: Log-Magnitude Plot');
set(gca,'xtick',[0,ws,wp,pi]/pi,'xgrid','on');
set(gca,'ytick',[-Asd,0],'ygrid','on');
print -deps2 ../EPSFILES/P0726
```

The filter response plots are shown in Figure 7.18.

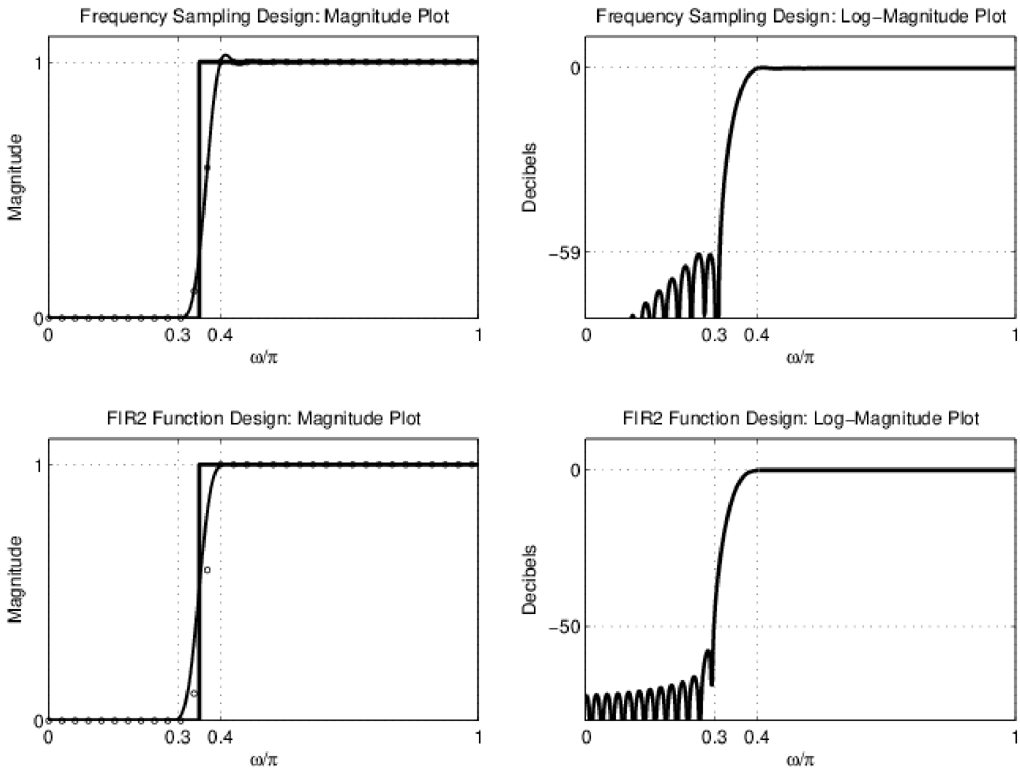


Figure 7.18: Filter design plots in Problem 7.26

P7.27

Consider the filter specifications given in Figure P7.1. Use the **fir2** function and a Hamming window to design a linear-phase FIR filter via the frequency sampling method. Experiment with the filter length to achieve the required design. Plot the amplitude response of the resulting filter.

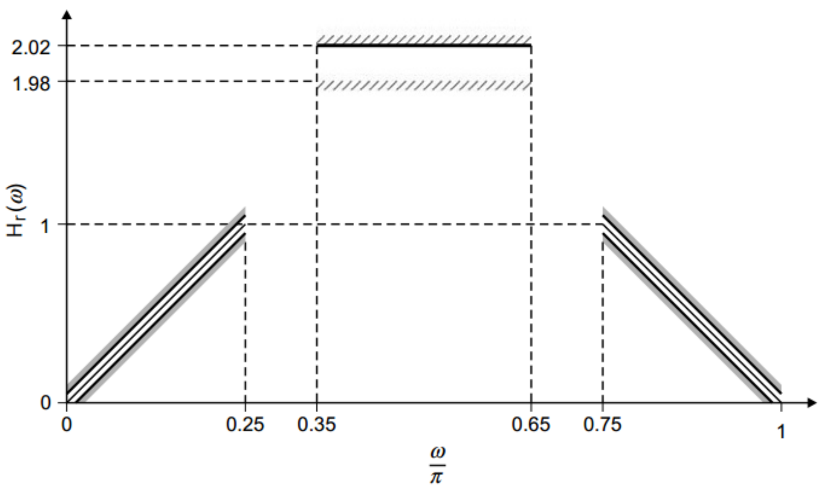


FIGURE P7.1 *Filter Specifications for Problem P7.27*

```
% P7.27
% Frequency sampling design using the fir2 function and a
Hamming window.
clear;clc; close all;
% Specifications:
fdr = [0,0.25,0.35,0.65,0.75,1]; Hdr = [0,1,2,2,1,0];
h = fir2(211,fdr,Hdr); [Hr,w,P,L] = ampl_res(h);
Hf_1 = figure('Units','inches','position',[1,1,5,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.27');
plot(w/pi,Hr,'r','linewidth',1.5); axis([0,1,-
0.05,2.05]);
xlabel('\omega/\pi'); ylabel('Amplitude');
title('FIR2 Function Design');
set(gca,'xtick',fdr,'xgrid','on','ytick',[0,1,1.98,2.02],
'ygrid','on');
print -deps2 ../EPSFILES/P0727
```

The filter amplitude response plot is shown in Figure 7.19.

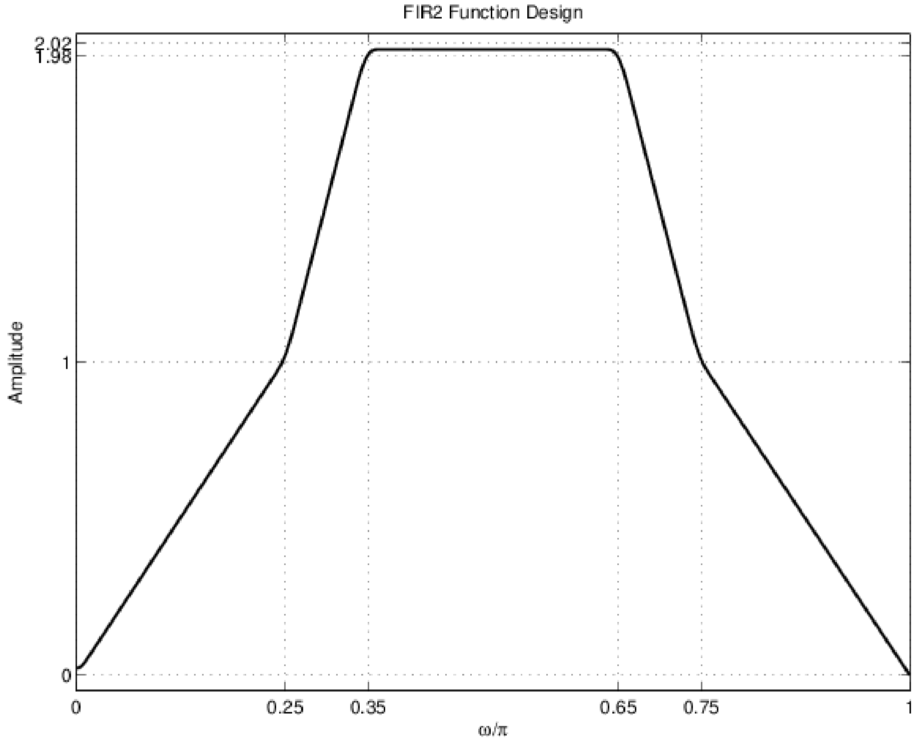


Figure 7.19: Filter design plots in Problem 7.27

P7.28

Design a bandpass filter using the frequency sampling method. Choose the order of the filter appropriately so that there is one sample in the transition band. Use optimum value for this sample. The specifications are as follows:

$$\left.\begin{array}{r}\text{lower stopband edge} = 0.3\pi \\ \text{upper stopband edge} = 0.7\pi\end{array}\right\} A_s \;=\; 40 \text{ dB}$$

$$\left.\begin{array}{r}\text{lower passband edge} = 0.4\pi \\ \text{upper passband edge} = 0.6\pi\end{array}\right\} R_p \;=\; 0.5 \text{ dB.}$$

Provide a plot of the log-magnitude response in dB and **stem** plot of the impulse response.

Solutions

```
% P7.28
clear;clc; close all;
% Specifications:
ws1 = 0.3*pi; % lower stopband edge
wp1 = 0.4*pi; % lower passband edge
wp2 = 0.6*pi; % upper passband edge
ws2 = 0.7*pi; % upper stopband edge
Rp = 0.5; % passband ripple
As = 40; % stopband attenuation
Hf_1 = figure('Units','inches','position',[1,1,7,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.28');
%% Frequency Sampling Design
% Choose M = 40 to get one sample in the transition band
M = 40; alpha = (M-1)/2; k = 0:M-1; wk = (2*pi/M)*k;
% Samples of the Frequency Response
T1 = 0.405;
Hrs =
[zeros(1,7),T1,ones(1,5),T1,zeros(1,13),T1,ones(1,5),T1,z
eros(1,6)];
% Ideal Amplitude Response for Plotting
fc1 = (wp1+ws1)/(2*pi); fc2 = (ws2+wp2)/(2*pi);
Hdr = [0,0,1,1,0,0]; fdr = [0,fc1,fc1,fc2,fc2,1];
% Compute the Impulse Response
k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
H = Hrs.*exp(j*angH); h = real(ifft(H,M)); n = 0:M-1;
% Compute Frequency Responses
```

```
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
% Actual Attn
Asd = floor(-max(db(ceil(ws2/delta_w)+2:end)));
% Filter Response Plots
subplot('position',[0.09,0.1,0.4,0.8]);
Hs_1 = stem(n,h,'g','filled'); set(Hs_1,'markersize',3);
axis([-1,M,-0.2,0.25]); xlabel('n'); ylabel('Amplitude');
title('Impulse Response'); set(gca,'xtick',[0,M-1]);
subplot('position',[0.59,0.1,0.4,0.8]);
plot(w/pi,db,'g','linewidth',2); axis([0,1,-80,10]);
xlabel('\omega/\pi'); ylabel('Decibels');
title('Log-Magnitude Plot');
set(gca,'xtick',[0,ws1,wp1,wp2,ws2,pi]/pi,'xgrid','on');
set(gca,'ytick',[-Asd,0],'ygrid','on');
print -deps2 ../EPSFILES/P0728
```

The filter response plots are shown in Figure 7.20.

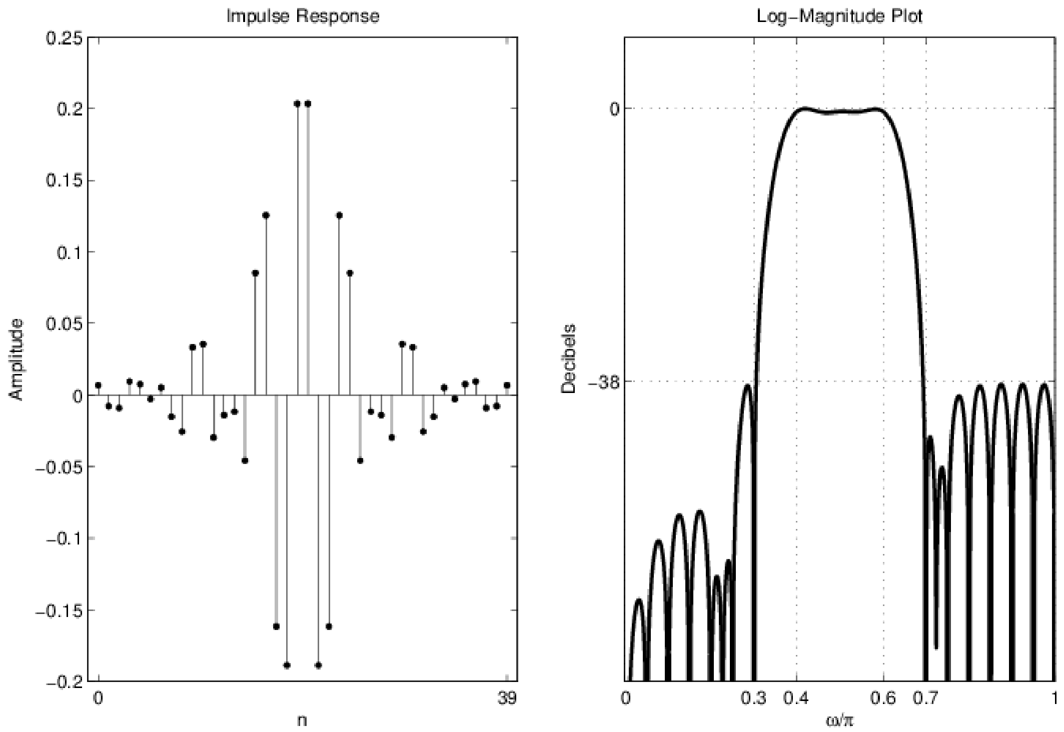


Figure 7.20: Filter design plots in Problem 7.28

P7.29

The frequency response of an ideal bandpass filter is given by

$$H_d(e^{j\omega}) = \begin{cases} 0, & 0 \leq |\omega| \leq \pi/3 \\ 1, & \pi/3 \leq |\omega| \leq 2\pi/3 \\ 0, & 2\pi/3 \leq |\omega| \leq \pi \end{cases}$$

1. Determine the coefficients of a 25-tap filter based on the Parks-McClellan algorithm with

stopband attenuation of 50 dB. The designed filter should have the smallest possible transition width.
2. Plot the amplitude response of the filter using the function developed in Problem P7.6.

Solutions

```
% P7.29
clear;clc;close all;
%% P0729a.m
clc; close all;
%% Specifications
wc1 = pi/3; % lower cutoff frequency
wc2 = 2*pi/3; % upper cuoff frequency
As = 50; % stopband attenuation
M = 25; % filter length
%
% (a) Design
tr_width = 2*pi*(As-13)/(14.36*(M-1)), % transition width
in radians
ws1 = wc1-tr_width/2; wp1 = wc1+tr_width/2;
wp2 = wc2-tr_width/2; ws2 = wc2+tr_width/2;
f = [0,ws1/pi,wp1/pi,wp2/pi,ws2/pi,1];
m = [0,0,1,1,0,0];
n = 0:M-1;
% h = remez(M-1,f,m);
h = firpm(M-1,f,m);
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)+1]))), % Actual
Attn
Rpd = -
min(db(ceil(wp1/delta_w)+1:floor(wp2/delta_w)+1)), %
Actial ripple
% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.29a');
subplot(2,1,1); Hs_1 = stem(n,h,'g','filled');
set(Hs_1,'markersize',3);
title('Impulse Response: Bandpass'); axis([-1,M,min(h)-
0.1,max(h)+0.1]);
xlabel('n','fontsize',10); ylabel('h(n)','fontsize',10);
set(gca,'XTick',[0;12;24]);
```

```
subplot(2,1,2); plot(w/pi,db,'g','linewidth',1.5);
title('Magnitude Response in dB','fontsize',10);
axis([0,1,-80,5]);
xlabel('\omega/\pi','fontsize',10);
ylabel('Decibels','fontsize',10);
set(gca,'XTick',f,'fontsize',10,'YTick',[-50;0]);
set(gca,'YTickLabel',{'-50';' 0 '});grid
print -deps2 ../EPSFILES/P0729a
```

The impulse and log-magnitude response plots are shown in Figure 7.21.

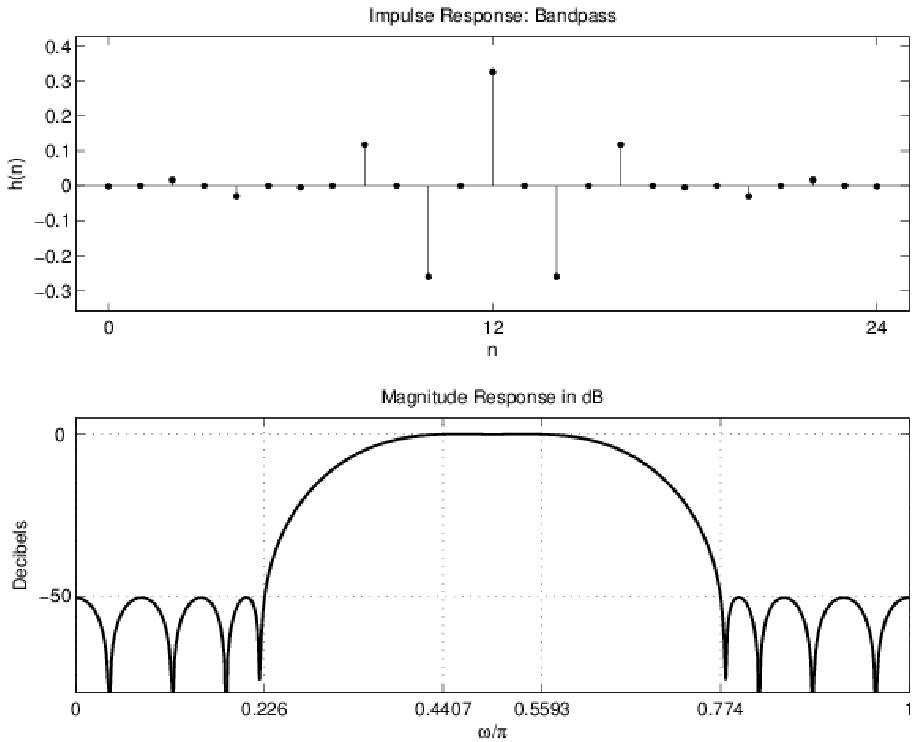


Figure 7.21: Filter impulse and log-magnitude response plots in Problem 7.29

```
%% P0729b.m
Hf_2 = figure('Units','inches','position',[1,1,6,3],...
'paperunits','inches');
set(Hf_2,'NumberTitle','off','Name','P7.29b');
[Hr,w,a,L] = Hr_Type1(h);
plot(w/pi,Hr,'g','linewidth',1.5);
title('Amplitude Response','fontsize',10)
xlabel('\omega/\pi','fontsize',10);
ylabel('Amplitude','fontsize',10);
set(gca,'XTickMode','manual','XTick',f);axis([0,1,-
0.1,1.1]);
print -deps2 ../EPSFILES/P0729b
```

The amplitude response plot is shown in Figure 7.22.

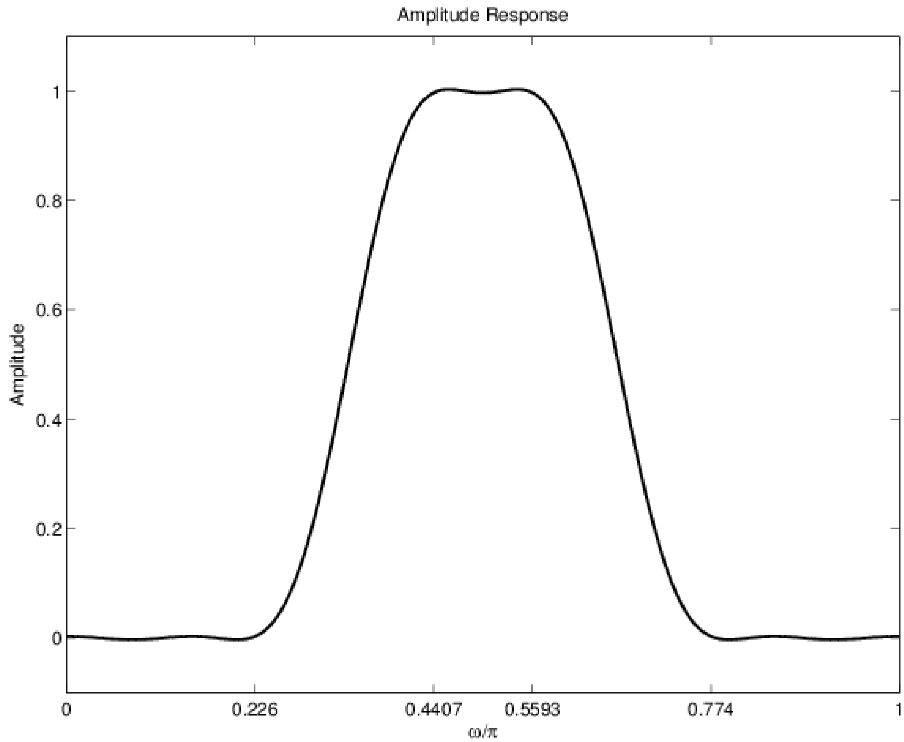


Figure 7.22: Filter amplitude response plot in Problem 7.29

P7.30

Consider the bandstop filter given in Problem P7.10.

1. Design a linear-phase bandstop FIR filter using the Parks-McClellan algorithm. Note that the length of the filter must be odd. Provide a plot of the impulse response and the magnitude response in dB of the designed filter.
2. Plot the amplitude response of the designed filter and count the total number of extrema in stopband and passbands. Verify this number with the theoretical estimate of the total number of extrema.
3. Compare the order of this filter with those of the filters in Problems P7.10 and P7.24.
4. Verify the operation of the designed filter on the following signal

$$x(n) = 5 - 5\cos\left(\frac{\pi n}{2}\right); \quad 0 \le n \le 300$$

Solutions

(a) Design using the Parks-McClellan algorithm and a plots of the impulse response and the magnitude response in dB of the designed filter.

```
% P7.30
%% P0730a.m
clc; close all;
%% Specifications
wp1 = 0.3*pi; % lower passband edge
```

```
ws1 = 0.4*pi; % lower stopband edge
ws2 = 0.6*pi; % upper stopband edge
wp2 = 0.7*pi; % upper passband edge
Rp = 0.25; % passband ripple
As = 50; % stopband attenuation
%
% 1. Design
delta1 = (10^(Rp/20)-1)/(10^(Rp/20)+1);
delta2 = (1+delta1)*(10^(-As/20));
weights = [delta2/delta1, 1, delta2/delta1];
delta_f =min((wp2-ws2)/(2*pi), (ws1-wp1)/(2*pi));
M = ceil((-20*log10(sqrt(delta1*delta2))-
13)/(14.6*delta_f)+1);

M = 2*floor(M/2)+1
f = [0, wp1/pi, ws1/pi, ws2/pi, wp2/pi, 1];
m = [1 1 0 0 1 1];
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,[1]);
delta_w = pi/500;
Asd = floor(-
max(db([floor(ws1/delta_w)+1:floor(ws2/delta_w)]))), %
Actual Attn
while Asd <50
M = M+2
h = remez(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,[1]);
delta_w = pi/500;
Asd = floor(-
max(db([floor(ws1/delta_w)+1:floor(ws2/delta_w)]))), %
Actual Attn
end
n = 0:M-1;
% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.30a');
subplot(2,1,1); Hs_1 = stem(n,h,'g','filled');
set(Hs_1,'markersize',3);
title('Impulse Response: Bandpass','fontsize',10);
axis([-1,M,min(h)-0.1,max(h)+0.1]);
xlabel('n','fontsize',10); ylabel('h(n)','fontsize',10)
set(gca,'XTickMode','manual','XTick',[0;17;34])
subplot(2,1,2); plot(w/pi,db,'g','linewidth',1.5);
```

```
title('Magnitude Response in dB','fontsize',10);
axis([0,1,-60,5]);
xlabel('\omega/\pi','fontsize',10);
ylabel('Decibels','fontsize',10)
set(gca,'XTickMode','manual','XTick',f);
set(gca,'YTickMode','manual','YTick',[-50;0])
set(gca,'YTickLabel',{' 50';' 0 '});grid;
print -deps2 ../EPSFILES/P0730a
```

The filter response plots are shown in Figure 7.23.

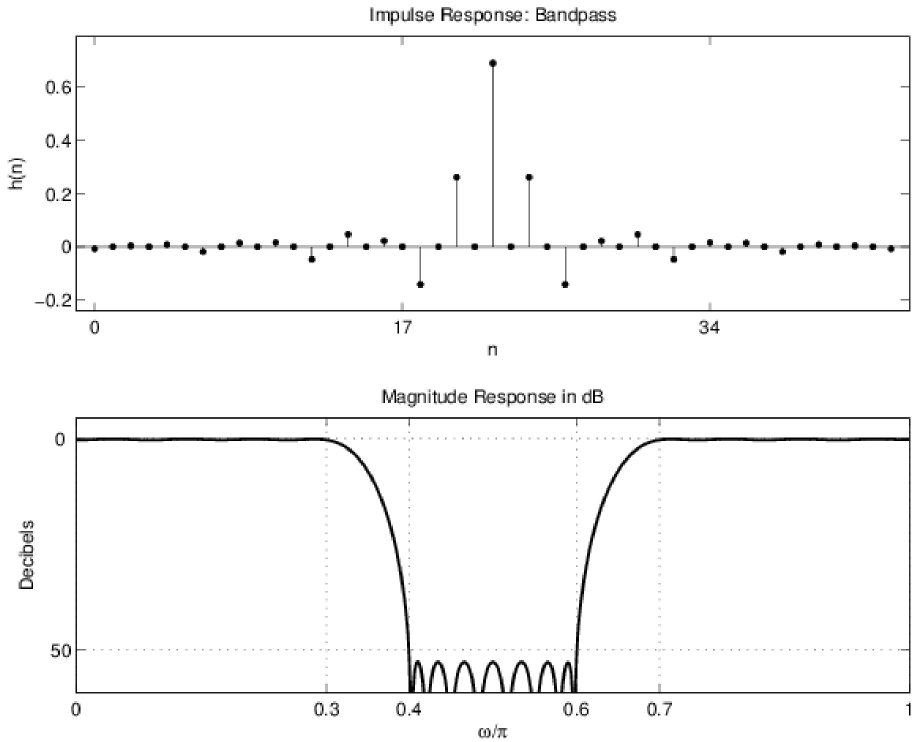


Figure 7.23: Filter impulse and log-magnitude response plots in Problem 7.30

(b) Plot of the amplitude response of the designed filter.

```
%% P0730b.m
Hf_2 = figure('Units','inches','position',[1,1,6,3],...
'paperunits','inches');
set(Hf_2,'NumberTitle','off','Name','P7.30b');
[Hr,w,a,L] = Hr_Type1(h);
plot(w/pi,Hr,'g','linewidth',1.5);
title('Amplitude Response','fontsize',10)
xlabel('\omega/\pi','fontsize',10);
ylabel('Amplitude','fontsize',10)
axis([0,1,-0.1,1.1]);
set(gca,'XTickMode','manual','XTick',f);
print -deps2 ../EPSFILES/P0730b
```

The amplitude response plot is shown in Figure 7.24.

From Figure 7.24 the total number of extrema in stopband and passbands are equal to 25. Since M = 45, L = 22. Then the extrema are L + 2 or L + 3. Hence this is a L + 32 = 25 equiripple design.

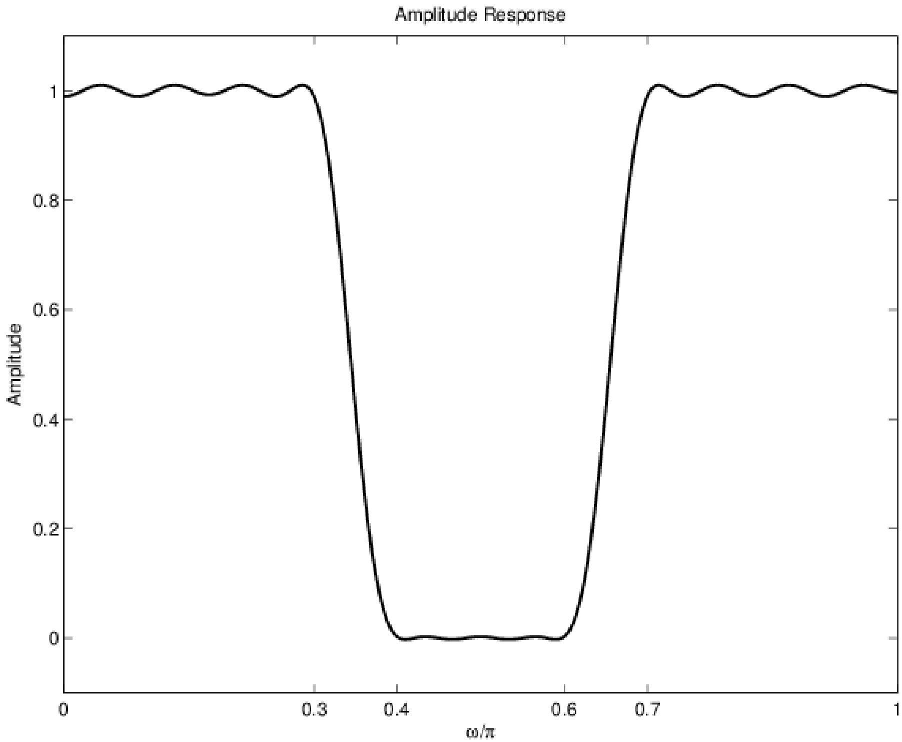


Figure 7.24: Filter amplitude response plot in Problem 7.30

(c) The filter order in Problem P7.10 is 67 - 1 = 66 while that in P7.14 is 49 - 1 = 48. The order using Parks-McClellan algorithm is 44.

(d) Operation of the designed filter on the following signal

$$x(n) = 5 - 5\cos\left(\frac{\pi n}{2}\right); \quad 0 \le n \le 300$$

```
%% P0730d
n = 0:300; x = 5 - 5*cos(pi*n/2); y = filter(h,1,x);
Hf_3 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_3,'NumberTitle','off','Name','P7.30d');
subplot(2,1,1); Hs_3 =
stem(n(1:101),x(1:101),'g','filled');
axis([0,100,-1,11]); xlabel('n','fontsize',10);
set(Hs_3,'markersize',3);
ylabel('x(n)','fontsize',10); title('Input
Signal','fontsize',10);
subplot(2,1,2); Hs_4 =
stem(n(1:101),y(1:101),'g','filled');
axis([0,100,-1,11]); xlabel('n','fontsize',10);
set(Hs_4,'markersize',3);
ylabel('y(n)','fontsize',10); title('Output
```

```
Signal','fontsize',10);
print -deps2 ../EPSFILES/P0730d
```

The input and output signal plots are shown in Figure 7.25.

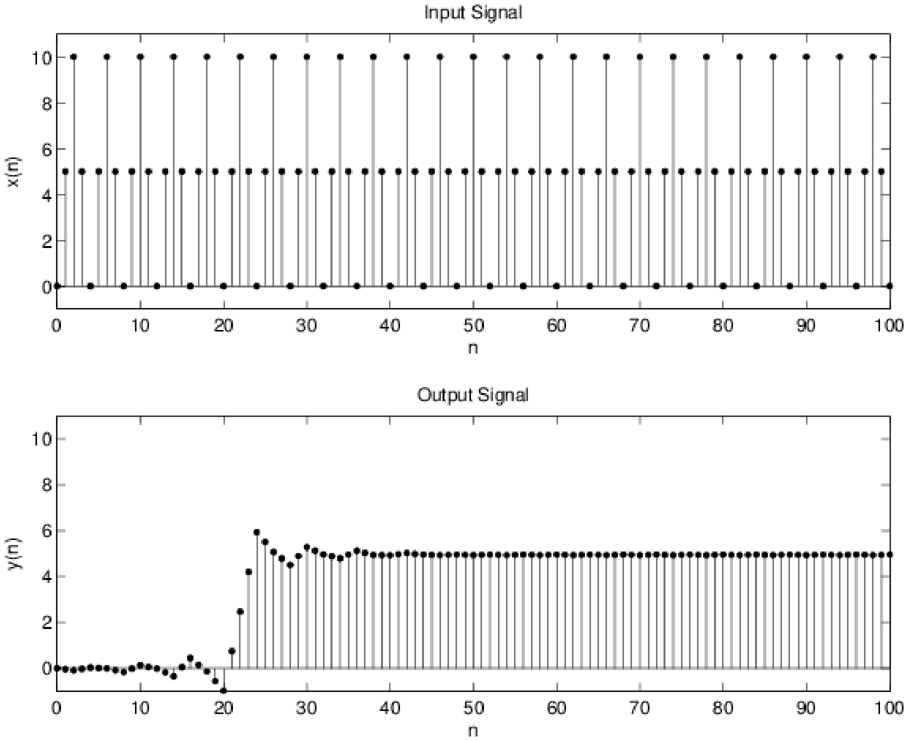


Figure 7.25: Input/output signal plots in Problem 7.30

P7.31

Using the Parks-McClellan algorithm, design a 25-tap FIR differentiator with slope equal to 1 sample/cycle.

1. Choose the frequency band of interest between 0.1π and 0.9π. Plot the impulse response and the amplitude response.
2. Generate 100 samples of the sinusoid

$$x(n) = 3\sin(0.25\pi n), \quad n = 0, ..., 100$$

and process through the preceding FIR differentiator. Compare the result with the theoretical "derivative" of $x(n)$. *Note:* Don't forget to take the 12-sample delay of the FIR filter into account.

Solutions

(a) Frequency band of interest between 0.1π and 0.9π and plot of the impulse response and the amplitude response:

```
% P7.31
%% P0731a.m
clc; close all;
```

```
%% Specifications
M = 25; w1 = 0.1*pi; w2 = 0.9*pi; % slope = 1 sam/cycle
%
% (a) Design
f = [w1/pi w2/pi]; m = [w1/(2*pi) w2/(2*pi)];
h = firpm(M-1,f,m,'differentiator'); [db,mag,pha,grd,w] =
freqz_m(h,1);
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.31a');
subplot(2,1,1); Hs_1 = stem([0:1:M-1],h,'g','filled');
set(Hs_1,'markersize',3); axis([-1,25,-0.2,.2]);
title('Impulse Response','fontsize',10);
xlabel('n','fontsize',10); ylabel('h(n)','fontsize',10);
set(gca,'XTick',[0;12;24],'fontsize',10);
subplot(2,1,2); [Hr,w,P,L] = ampl_res(h);
plot(w/(2*pi), Hr,'g','linewidth',1.5);
title('Amplitude Response','fontsize',10); grid;
axis([0,0.5,0,0.5]);
set(gca,'XTick',[0;w1/(2*pi);w2/(2*pi);0.5]);
set(gca,'YTick',[0;0.5]);
print -deps2 ../EPSFILES/P0731a
```

The filter impulse and amplitude response plots are shown in Figure 7.26.

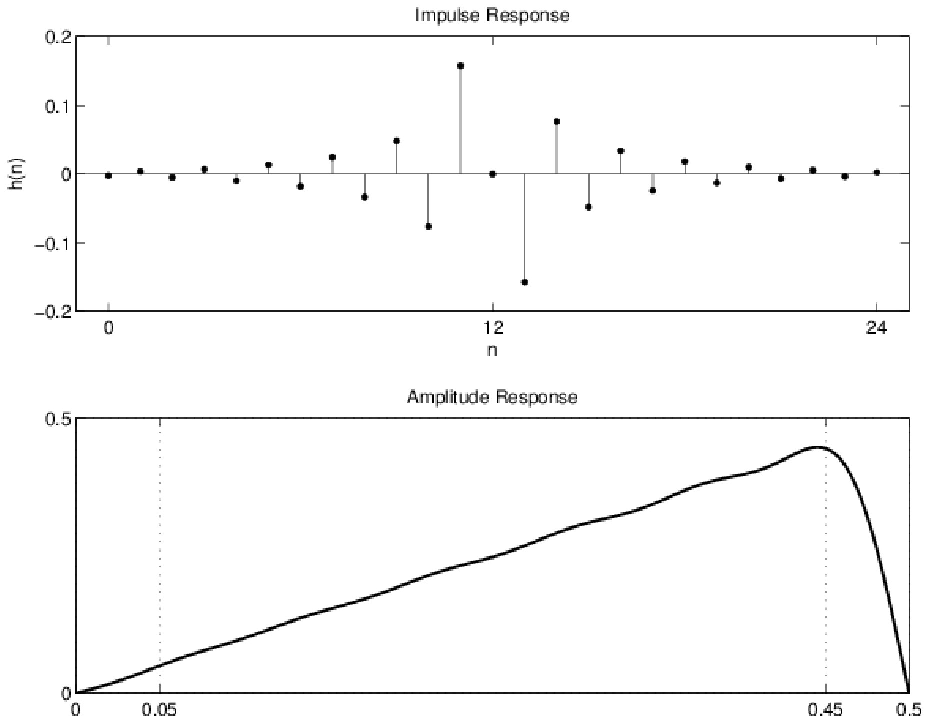


Figure 7.26: Filter impulse and amplitude response plots in Problem 7.31

(b) Verification of the filter performance using 100 samples of the sinusoid

$$x(n) = 3\sin(0.25\pi n), \quad n = 0, ..., 100$$

```
%% (b) Differentiator verification
Hf_1 = figure('Units','inches','position',[1,1,6,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.31b');
n=[0:1:100]; x = 3*sin(0.25*pi*n); y = conv(x,-h); m =
[41:1:81];
plot(m,x(41:1:81),'g',m,y(41+12:1:81+12),'m',...
'linewidth',1.5); grid; % add 12 sample delay to y
xlabel('n','fontsize',10); title('Input-Output
Sequences','fontsize',10);
axis([40,82,-4,4]); set(gca,'XTick',[41;81]);
set(gca,'YTick',[-3;0;3]);
ylabel('Amplitude','fontsize',10);
print -deps2 ../EPSFILES/P0731b
```

The filter input-output plots are shown in Figure 7.27. Since the slope is $\pi/2$ sam/rad, the gain at $\omega = 0.25\pi$ is equal to 0.125. Therefore, the output (when properly shifted) should be

$$y(n) = 3(0.125)\cos(0.25\pi n) = 0.375\cos(0.25\pi n)$$

From the Figure 7.27 we can verify that $y(n)$(the lower curve) is indeed a cosine waveform with amplitude ≈ 0.4.

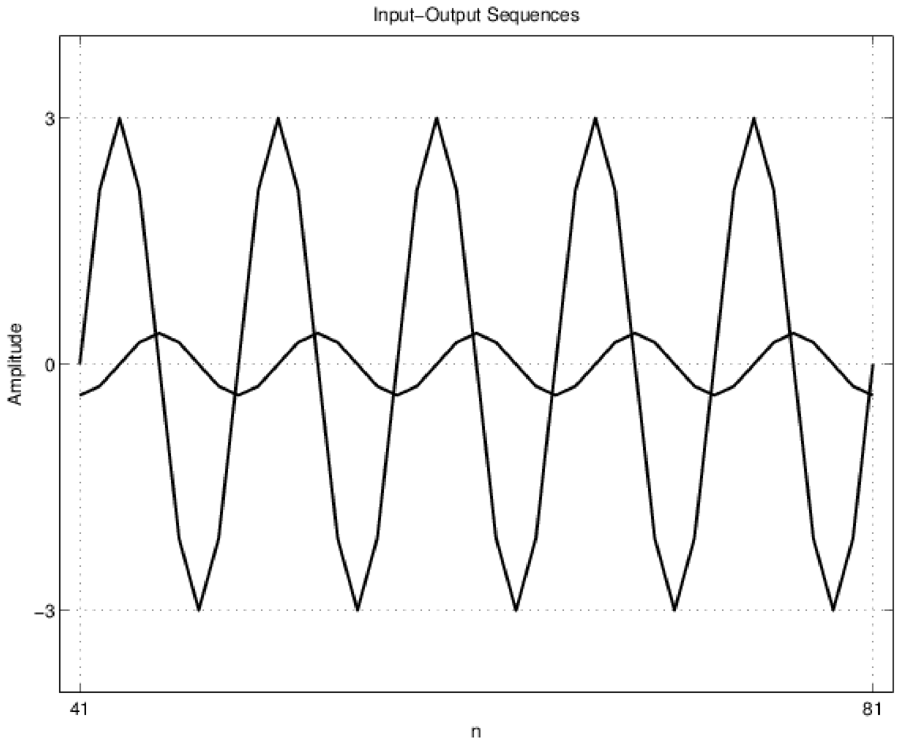


Figure 7.27: Filter input-output plots in Problem 7.31

P7.32

Design a lowest-order equiripple linear-phase FIR filter to satisfy the specifications given in Figure P7.2. Provide a plot of the amplitude response and a plot of the impulse response.

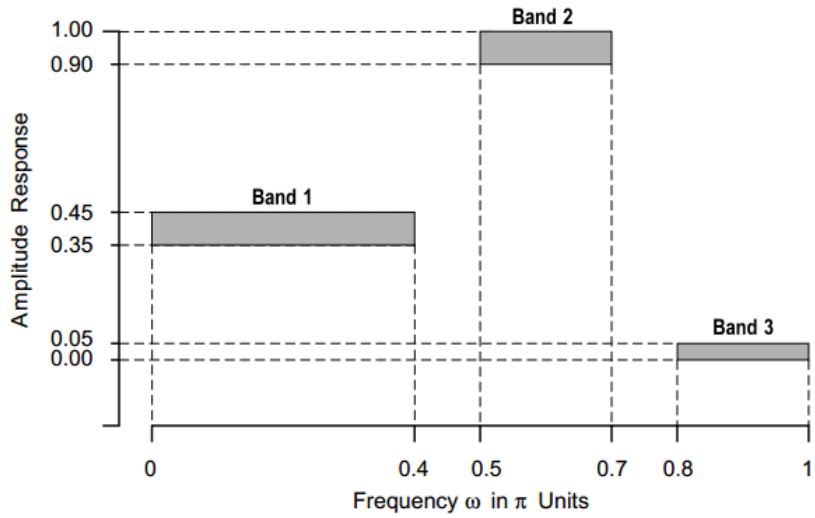


FIGURE P7.2 *Filter Specifications for Problem P7.32*

Solutions

```
% P7.32
clear;clc; close all;
%% Specifications
f = [0,0.4,0.5,0.7,0.8,1]; m =
[0.4,0.4,0.95,0.95,0.025,0.025];
delta1 = 0.05; delta2 = 0.05; delta3 = 0.025;
weights = [delta3/delta1, delta3/delta2, delta3/delta3];
As = -20*log10(0.05)
% Design
delta_f = 0.05; % Transition width in cycles per sample
M = ceil((-20*log10(sqrt(delta2*delta3))-
13)/(14.6*delta_f)+1)
h = firpm(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([(0.8*pi/delta_w)+1:501]))), % Actual
Attn
while Asd<floor(As)
M = M+2
h = firpm(M-1,f,m,weights);
[db,mag,pha,grd,w] = freqz_m(h,1);
```

```
Asd = floor(-max(db([(0.8*pi/delta_w)+1:501]))), % Actual
Attn
end
% M = M+2
% h = firpm(M-1,f,m,weights);
% [db,mag,pha,grd,w] = freqz_m(h,1);
% Asd = floor(-max(db([(0.8*pi/delta_w)+1:501]))), %
Actual Attn
n = 0:M-1;
% Filter Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.32');
subplot(2,1,1); Hs_1 = stem(n,h,'g','filled');
set(Hs_1,'markersize',3);
title('Impulse Response','fontsize',10);
axis([-1,M,min(h)-0.1,max(h)+0.1]);
xlabel('n','fontsize',10);
ylabel('h(n)','fontsize',10); set(gca,'XTick',[0;13;26])
% Amplitude Response Plot
[Hr,w,a,L] = Hr_Type1(h);
subplot(2,1,2); plot(w/pi,Hr,'g','linewidth',1.5);
title('Amplitude Response','fontsize',10);
axis([0,1,0,1])
xlabel('\omega/\pi','fontsize',10);
ylabel('Hr(w)','fontsize',10)
set(gca,'XTick',f,'YTick',[0;0.05;0.35;0.45;0.9;1]); grid
print -deps2 ../EPSFILES/P0732

As =
   26.0206
M =
    23
Asd =
    24
M =
    25
Asd =
    25
M =
    27
Asd =
    26
```

The filter amplitude and impulse response plots are shown in Figure 7.28.

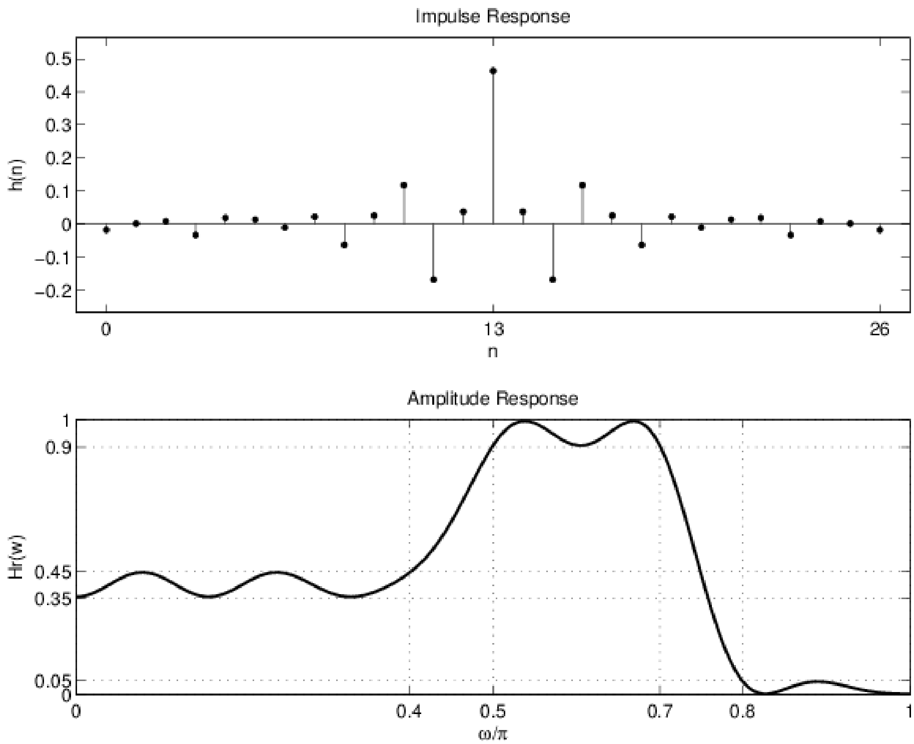


Figure 7.28: Filter amplitude and impulse response plots in Problem 7.32

P7.33

A digital signal $x(n)$ contains a sinusoid of frequency $\pi/2$ and a Gaussian noise $w(n)$ of zero mean and unit variance; i.e.,

$$x(n) = 2\cos\frac{\pi n}{2} + w(n)$$

We want to filter out the noise component using a 50th-order causal and linear-phase FIR filter.
1. Using the Parks-McClellan algorithm, design a narrow bandpass filter with passband width of no more than 0.02π and stopband attenuation of at least 30 dB. Note that no other parameters are given and that you have to choose the remaining parameters for the **firpm** function to satisfy the requirements. Provide a plot of the log-magnitude response in dB of the designed filter.
2. Generate 200 samples of the sequence $x(n)$ and processed through the preceding filter to obtain the output $y(n)$. Provide subplots of $x(n)$ and $y(n)$ for $100 \le n \le 200$ on one plot and comment on your results.

Solutions

(a) In this design we already know the order of the filter. The only parameters that we don't know are the stopband cutoff frequencies ω_{s1} and ω_{s2}. Let the transition bandwidth be $\Delta\omega$ and let the passband be symmetrical with respect to the center frequency ω_c. Then

$$\omega_{p_1} = \omega_c - 0.01\pi,\ \omega_{p_2} = \omega_c + 0.01\pi,\ \omega_{s_1} = \omega_{p_1} - \Delta\omega,\ \text{and}\ \omega_{s_2} = \omega_{p_2} + \Delta\omega$$

We will also assume that the tolerances in each band are equal. Now we will begin with initial value for $\Delta\omega = 0.2\pi$ and run the firpm algorithm to obtain the actual stopband attenuation. If it is smaller (larger) than the given 30 dB then we will increase (decrease) $\Delta\omega$then iterate to obtain the desired solution. The desired solution was found for $\Delta\omega = 0.5\pi$. Matlab Script:

```
% P7.33
clear; close all;
%% P0733a.m
% Specifications
N = 50; % Order of the filter
w0 = 0.5*pi; % Center frequency
Bandwidth = 0.02*pi; % Bandwidth
%
% Deltaw = Transition bandwidth (iteration variable)
%
wp1 = w0-Bandwidth/2; wp2 = w0+Bandwidth/2;
% (a) Design
Deltaw = 0.02*pi; % Initial guess
ws1=wp1-Deltaw; ws2=wp2+Deltaw;
F=[0, ws1, wp1, wp2, ws2, pi]/pi;
m=[0,0,1,1,0,0];
% h=remez(50,F,m);
h = firpm(50,F,m);
[db,mag,pha,grd,w]=freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)]))), % Actual
Attn
while Asd < 30
    % Next iteration
Deltaw = Deltaw+0.01*pi;
ws1=wp1-Deltaw; ws2=wp2+Deltaw;
F=[0, ws1, wp1, wp2, ws2, pi]/pi;
h = firpm(50,F,m);
[db,mag,pha,grd,w]=freqz_m(h,1);
delta_w = pi/500;
Asd = floor(-max(db([1:floor(ws1/delta_w)]))), % Actual
Attn
end
Hf_1 = figure('Units','inches','position',[1,1,6,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.33a');
plot(w/pi,db,'g','linewidth',1.5); axis([0,1,-50,0]);
title('Log-Magnitude Response','fontsize',10);
xlabel('\omega/\pi','fontsize',10);
ylabel('DECIBELS','fontsize',10)
set(gca,'XTick',[0;ws1/pi;ws2/pi;1],'YTick',[-30;0])
```

```
set(gca,'YTickLabel',{' 30';' 0 '});grid
print -deps2 ../EPSFILES/P0733a

Asd =
    13
Asd =
    20
Asd =
    26
Asd =
    30
```

The log-magnitude response is shown in Figure 7.29.

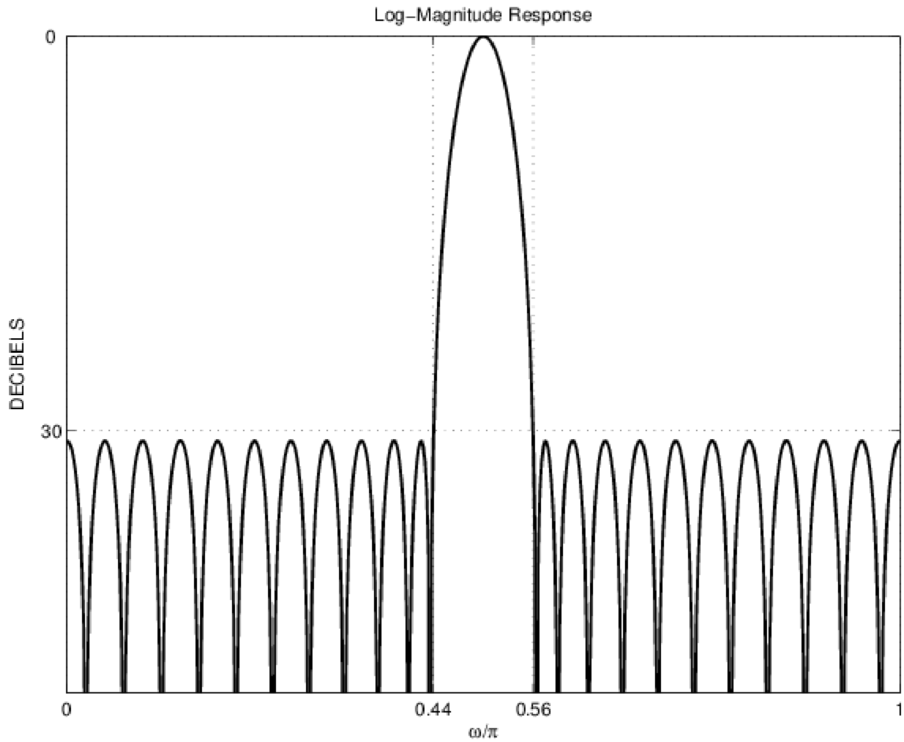


Figure 7.29: Log-Magnitude Response Plot in Problem P7.33a

(b) The time-domain response of the filter. Matlab script:

```
%% P0733b.m
% (b) Time-domain Response
n = [0:1:200]; x = 2*cos(pi*n/2)+randn(1,201); y =
filter(h,1,x);
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.33b');
subplot(2,1,1); Hs_1 =
stem(n(101:201),x(101:201),'g','filled');
title('Input sequence x(n)','fontsize',10);
set(Hs_1,'markersize',3);
ylabel('Amplitude','fontsize',10);
```

```
subplot(2,1,2); Hs_2 =
stem(n(101:201),y(101:201),'m','filled');
title('Output sequence y(n)','fontsize',10);
set(Hs_2,'markersize',3);
xlabel('n','fontsize',10);
ylabel('Amplitude','fontsize',10);
print -deps2 ../EPSFILES/P0733b
```

The time-domain response is shown in Figure 7.30.

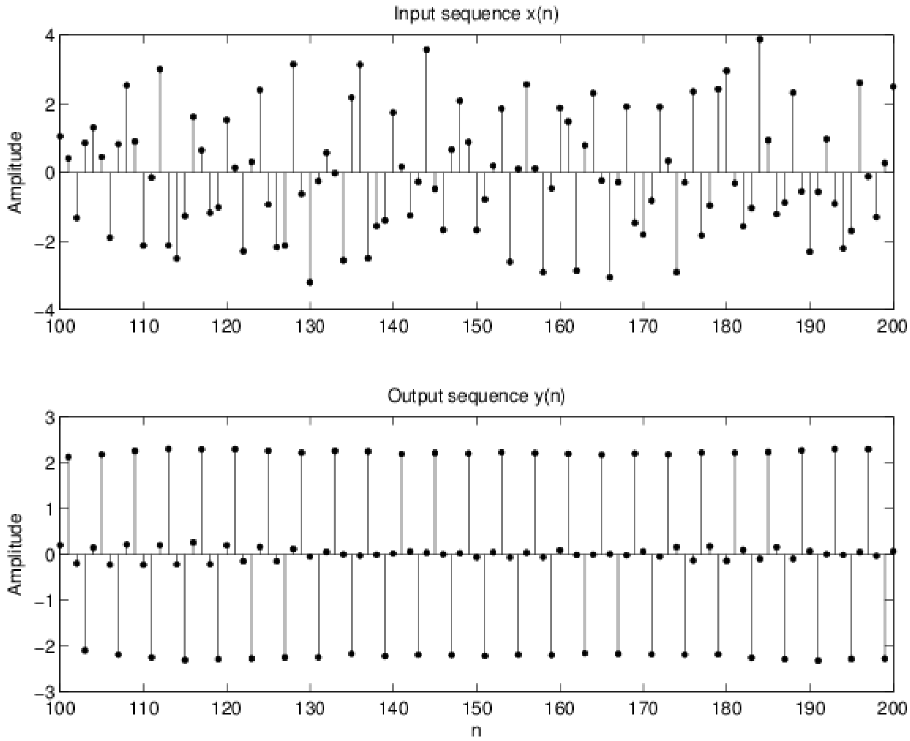


Figure 7.30: The Time-domain Response in Problem P7.33b

P7.34

Design a minimum order linear-phase FIR filter, using the Parks-McClellan algorithm, to satisfy the requirements given in Figure P7.1.

1. Provide a plot of the amplitude response with grid-lines and axis labeling as shown in Figure P7.1.

2. Generate the following signals

$$x_1(n) = \cos(0.25\pi n), \quad x_2(n) = \cos(0.5\pi n), \quad x_3(n) = \cos(0.75\pi n); \quad 0 \le n \le 100.$$

Process these signals through this filter to obtain the corresponding output signals $y_1(n)$, $y_2(n)$, and $y_3(n)$. Provide stem plots of all input and output signals in one figure.

(a) Design and plot of the amplitude response with grid-lines and axis labeling as shown in Figure P7.1:

```
% P7.34
clear;clc; close all;
%% P0734a.m
% Specifications
f = [0,0.25,0.35,0.65,0.75,1]; m = [0,1,2,2,1,0];
% Optimum Design
h = firpm(46,f,m); [db,mag,pha,grd,w] = freqz_m(h,1);
Hf_1 = figure('Units','inches','position',[1,1,6,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.34a');
plot(w/pi,mag,'g','linewidth',1.5); axis([0,1,0,2.2]);
xlabel('\omega/\pi','fontsize',10);
ylabel('Amplitude','fontsize',10);
title('Amplitude Response','fontsize',10);
set(gca,'xtick',f,'ytick',[1,2]); grid;
print -deps2 ../EPSFILES/P0734a
```

The amplitude response is shown in Figure 7.31.

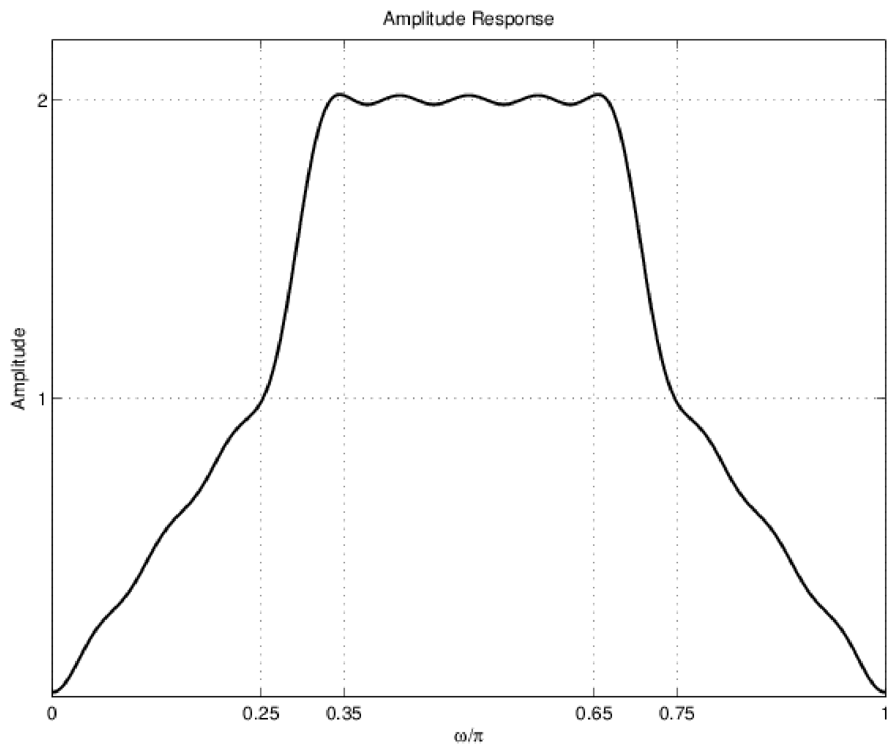


Figure 7.31: Amplitude Response Plot in Problem P7.34a

(b) Verification of the filter performance using the following signals

$$x_1(n) = \cos(0.25\pi n), \quad x_2(n) = \cos(0.5\pi n), \quad x_3(n) = \cos(0.75\pi n); \quad 0 \le n \le 100.$$

```
%% P0734b.m
% Input/Output Responses
n = 0:100; x =
1*cos(0.25*pi*n)+0*cos(0.5*pi*n)+cos(pi*n);
y = filter(h,1,x);
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.34b');
subplot(2,1,1); Hs_1 = stem(n,x,'g','filled');
set(Hs_1,'markersize',3);
title('Input Sequence','fontsize',10);
ylabel('x(n)','fontsize',10);
subplot(2,1,2); Hs_2 = stem(n,y,'m','filled');
set(Hs_2,'markersize',3);
title('Output Sequence','fontsize',10);
ylabel('y(n)','fontsize',10);
xlabel('n','fontsize',10);
print -deps2 ../EPSFILES/P0734b
```

The time-domain response is shown in Figure 7.32.

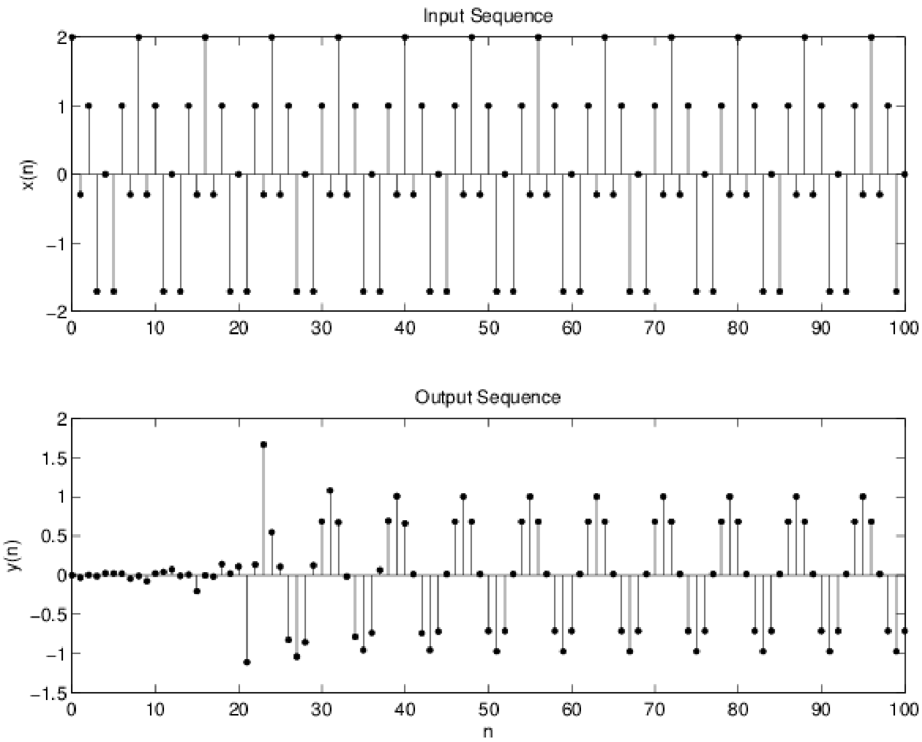


Figure 7.32: The Time-domain Response in Problem P7.34b

P7.35

Design a minimum-order linear-phase FIR filter, using the Parks-McClellan algorithm, to satisfy the requirements given in Figure P7.3. Provide a plot of the amplitude response with grid-lines and axis labeling as shown in Figure P7.3.

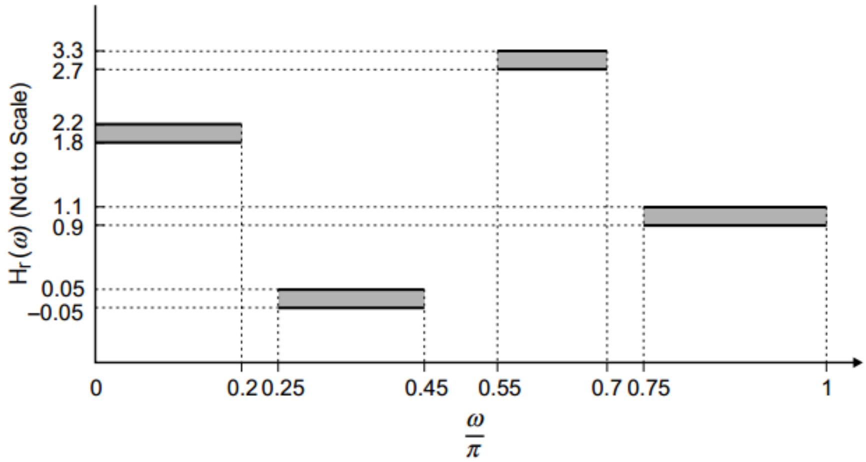


FIGURE P7.3 *Filter Specifications for Problem P7.35*

Solutions

Design of a minimum order linear-phase FIR filter, using the Parks-McClellan algorithm, to satisfy the requirements given in Figure P7.3 and a plot of the amplitude response with grid-lines and axis labeling as shown in Figure P7.3.

```
% P7.35
clear;clc; close all;
%Specifications
f = [0,0.2,0.25,0.45,0.55,0.7,0.75,1]; m =
[2,2,0,0,3,3,1,1];
delta = [0.2,0.05,0.3,0.1]; YT = [1;-1]*delta; YT =
(YT(:))'; YT = m+YT;
weight = delta(2)*ones(1,4)./delta;
% Optimum Design
M = 45;h = firpm(M-1,f,m,weight); [Hr,w,a,L] =
Hr_Type1(h); Hr_min = -min(Hr(126:226));
disp(sprintf('\n Achieved Tolerance in the stopband
= %5.2f',Hr_min));
Hf_1 = figure('Units','inches','position',[1,1,6,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.35');
plot(w/pi,Hr,'g','linewidth',1.5); axis([0,1,-0.1,3.5]);
hold on; m = YT;
```

```
plot([f(1),f(2)],[m(1),m(1)],'y:',[f(1),f(2)],[m(2),m(2)]
,'y:');
plot([f(3),f(4)],[m(3),m(3)],'g:',[f(3),f(4)],[m(4),m(4)]
,'g:');
plot([f(5),f(6)],[m(5),m(5)],'m:',[f(5),f(6)],[m(6),m(6)]
,'m:');
plot([f(7),f(8)],[m(7),m(7)],'c:',[f(7),f(8)],[m(8),m(8)]
,'c:');
title('Amplitude Response','fontsize',10)
xlabel('\omega/\pi','fontsize',10);ylabel('H_{r}(\omega)'
,'fontsize',10);
print -deps2 ../EPSFILES/P0735
set(gca,'XTickMode','manual','XTick',f);
set(gca,'YTickMode','manual','YTick',sort(YT));
hold off;%grid;
```

The amplitude response is shown in Figure 7.33.

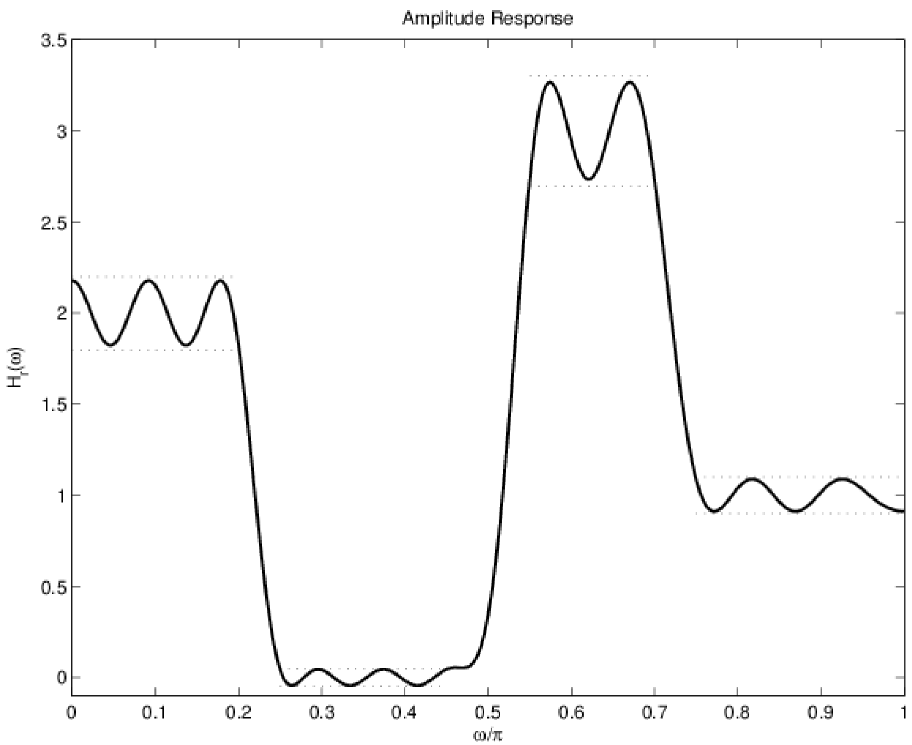


Figure 7.33: Amplitude Response Plot in Problem P7.35

P7.36

The specifications on the amplitude response of an FIR filter are given in Figure P7.4.

1. Using a window design approach and a *fixed* window function, design a minimum-length linear-phase FIR filter to satisfy the given requirements. Provide a plot of the amplitude response with grid-lines as shown in Figure P7.4.

2. Using a window design approach and the Kaiser window function, design a minimum-length linear-phase FIR filter to satisfy the given requirements. Provide a plot of the amplitude response with grid-lines as shown in Figure P7.4.
3. Using a frequency-sampling design approach and with no more than two samples in the transition bands, design a minimum-length linear-phase FIR filter to satisfy the given requirements. Provide a plot of the amplitude response with grid-lines as shown in Figure P7.4.
4. Using the Parks-McClellan design approach, design a minimum-length linear-phase FIR filter to satisfy the given requirements. Provide a plot of the amplitude response with grid-lines as shown in Figure P7.4.
5. Compare the preceding four design methods in terms of
• the order of the filter
• the exact band-edge frequencies
• the exact tolerances in each band

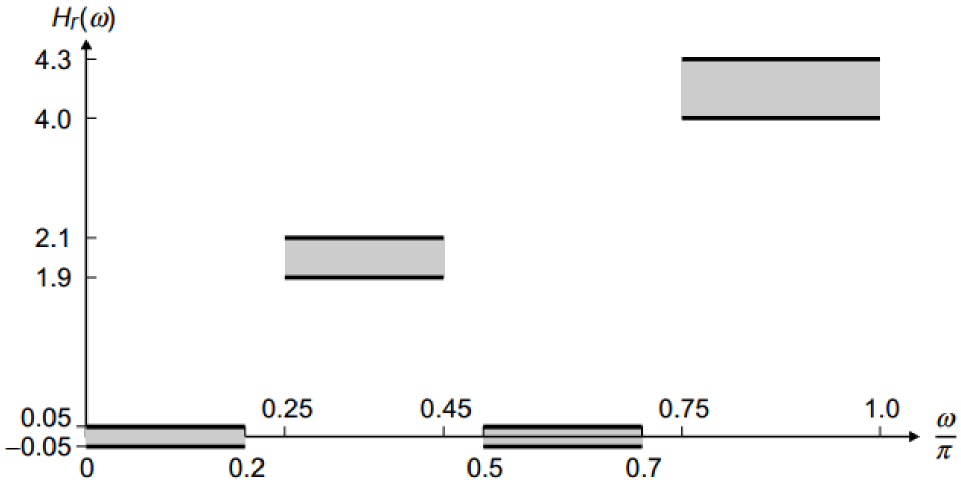


FIGURE P7.4 *Filter Specifications for Problem P7.36*

Solutions

The specifications on the amplitude response of an FIR filter are given in Figure P7.4.

```
% P7.36
clear;clc; close all;
% Given specifications
f = [0,0.2,0.25,0.45,0.5,0.7,0.75,1]; % Bandedge
Frequencies
m = [0,0,2,2,0,0,4.15,4.15]; % Nominal (Ideal) band gains
d1 = 0.05; d2 = 0.1; d3 = 0.05; d4 = 0.15; % Band
tolerances
% Stopband Attenuation
As = ceil(-20*log10(d1/(m(7)+d4)))
% Band-edge indices
I1 = 500*f(1)+1; % index into omega array for f(1) band-
```

```
edge
I2 = 500*f(2)+1; % index into omega array for f(2) band-
edge
I3 = 500*f(3)+1; % index into omega array for f(3) band-
edge
I4 = 500*f(4)+1; % index into omega array for f(4) band-
edge
I5 = 500*f(5)+1; % index into omega array for f(5) band-
edge
I6 = 500*f(6)+1; % index into omega array for f(6) band-
edge
I7 = 500*f(7)+1; % index into omega array for f(7) band-
edge
I8 = 500*f(8)+1; % index into omega array for f(8) band-
edge
```

(a) Minimum-length linear-phase FIR filter design using a *fixed* window function, to satisfy the given requirements and a plot of the amplitude response with grid-lines as shown in Figure P7.4:

```
%% P0736a.m
% (a) Fixed window design: Stopband Attn <= 39db =>
Hanning Window
delta_f = f(3)-f(2); M = 6.2/delta_f; M = floor(M/2)*2+1;
% The above value of M is 125. The minimum M was found to
be 119
M = 119; N = M-1;
w_han = hanning(M)';
h_ideal = m(7)*(ideal_lp(pi,M) - ideal_lp(0.725*pi,M))...
+ m(3)*(ideal_lp(0.475*pi,M) - ideal_lp(0.225*pi,M));
h = h_ideal.*w_han;
[Hr,w,a,L] = Hr_Type1(h);
% Computation of Exact band edges
w1 = 0;
w2 = w(max(find(Hr(I2:I3) <= 0.05)) + I2-1)/pi;
w3 = w(min(find(Hr(I2:I3) >= 1.90)) + I2-1)/pi;
w4 = w(max(find(Hr(I4:I5) >= 1.90)) + I4-1)/pi;
w5 = w(min(find(Hr(I4:I5) <= 0.05)) + I4-1)/pi;
w6 = w(max(find(Hr(I6:I7) <= 0.05)) + I6-1)/pi;
w7 = w(min(find(Hr(I6:I7) >= 4.00)) + I6-1)/pi;
w8 = 1;
% Computation of Exact tolerances
m1 = abs(min(Hr(I1:I2)));
m2 = max(Hr(I3:I4))-m(3);
m3 = abs(min(Hr(I5:I6)));
```

```
m4 = max(Hr(I7:I8))-m(7);
%
% Plot of Amplitude Response
Hf_1 = figure('paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.36a');
plot(w/pi,Hr,'g','linewidth',1.5); set(gca,'fontsize',8);
axis([0,1,-0.05,4.3]);
title('Amplitude Response in Part 1','fontsize',10);
xlabel('\omega/\pi','fontsize',10);
ylabel('Hr(w)','fontsize',10);grid;
set(gca,'XTickMode','manual','XTick',f);
set(gca,'YTickMode','manual','YTick',[-
0.05;0.05;1.9;2.1;4.0;4.3]);
% Printout of Order, Exact bandedges, and Exact
tolerances
disp(sprintf('\n(a) Fixed Window Design:'));
disp(sprintf(' Order : %2i',N));
disp(sprintf(' Exact band-
edges: %5.4f, %5.4f, %5.4f, %5.4f, %5.4f, %5.4f',...
w2, w3, w4, w5, w6, w7));
disp(sprintf(' Exact
tolerances: %5.4f, %5.4f, %5.4f, %5.4f,',...
m1, m2, m3, m4));
print -deps2 ../EPSFILES/P0736a
```

A plot of the amplitude response with grid-lines and axis labeling is shown in Figure 7.34.

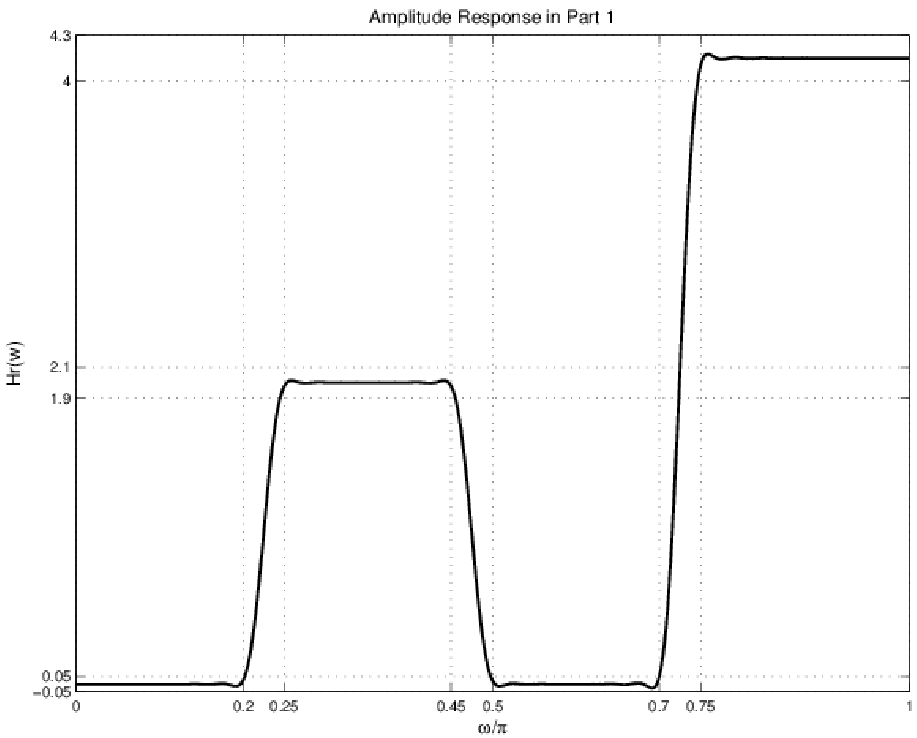


Figure 7.34: Amplitude Response Plot in Problem P7.36a

(b) Minimum-length linear-phase FIR filter design using the Kaiser window function to satisfy the given requirements and a plot of the amplitude response with grid-lines as shown in Figure P7.4:

```
%% P0736b.m
% (b) Kaiser Window Design:
delta_f = (f(3)-f(2))/2;
M = ceil((As - 7.95)/(14.36*delta_f) + 1); M =
floor(M/2)*2+1;
% The above value of M is 89. The minimum M was found to
be 89
M = 89; N = M-1;
beta = 0.5842*(As-21)^0.4 + 0.07886*(As-21);
w_kai = kaiser(M,beta)';
h_ideal = m(7)*(ideal_lp(pi,M) - ideal_lp(0.725*pi,M))...
+ m(3)*(ideal_lp(0.475*pi,M) - ideal_lp(0.225*pi,M));
h = h_ideal.*w_kai;
[Hr,w,a,L] = Hr_Type1(h);
% Computation of Exact band edges
w1 = 0;
w2 = w(max(find(Hr(I2:I3) <= 0.05)) + I2-1)/pi;
w3 = w(min(find(Hr(I2:I3) >= 1.90)) + I2-1)/pi;
w4 = w(max(find(Hr(I4:I5) >= 1.90)) + I4-1)/pi;
w5 = w(min(find(Hr(I4:I5) <= 0.05)) + I4-1)/pi;
w6 = w(max(find(Hr(I6:I7) <= 0.05)) + I6-1)/pi;
w7 = w(min(find(Hr(I6:I7) >= 4.00)) + I6-1)/pi;
w8 = 1;
% Computation of Exact tolerances
m1 = abs(min(Hr(I1:I2)));
m2 = max(Hr(I3:I4))-m(3);
m3 = abs(min(Hr(I5:I6)));
m4 = max(Hr(I7:I8))-m(7);
%
% Plot of Amplitude Response
Hf_2 = figure('paperunits','inches');
set(Hf_2,'NumberTitle','off','Name','P7.36b');
plot(w/pi,Hr,'g','linewidth',1.5); set(gca,'fontsize',8);
axis([0,1,-0.05,4.3]);
title('Amplitude Response in Part 2','fontsize',10);
xlabel('\omega/\pi','fontsize',10);
ylabel('Hr(w)','fontsize',10);grid;
set(gca,'XTickMode','manual','XTick',f);
set(gca,'YTickMode','manual','YTick',[-
0.05;0.05;1.9;2.1;4.0;4.3]);
```

```
%
% Printout of Order, Exact bandedges, and Exact
tolerances
disp(sprintf('\n(b) Kaiser Window Design:'));
disp(sprintf(' Order : %2i',N));
disp(sprintf(' Exact band-
edges: %5.4f, %5.4f, %5.4f, %5.4f, %5.4f, %5.4f',...
w2, w3, w4, w5, w6, w7));
disp(sprintf(' Exact
tolerances: %5.4f, %5.4f, %5.4f, %5.4f,',...
m1, m2, m3, m4));
print -deps2 ../EPSFILES/P0736b
```

A plot of the amplitude response with grid-lines and axis labeling is shown in Figure 7.35.

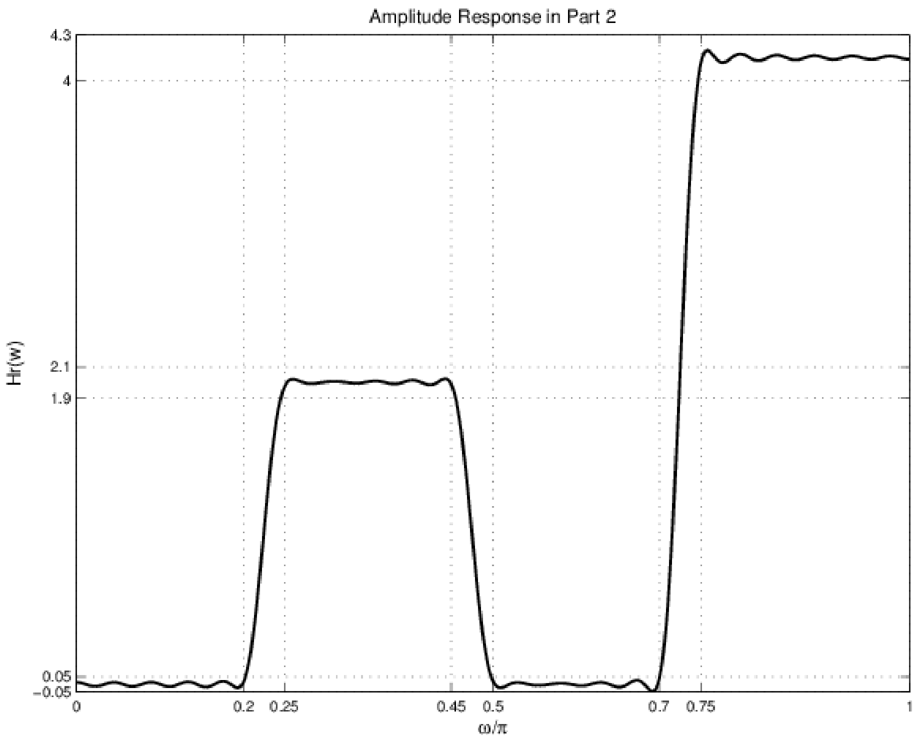


Figure 7.35: Amplitude Response Plot in Problem P7.36b

(c) Minimum-length linear-phase FIR filter design using a frequency-sampling design approach with no more than two optimum samples in the transition bands to satisfy the given requirements and a plot of the amplitude response with grid-lines as shown in Figure P7.4:

```
%% P0736c.m
% (c) Frequency Sampling Design: Choose M = 81
%
M = 81; N = M-1; alpha = (M-1)/2; l = 0:M-1; wl =
(2*pi/M)*l;
% The following parameters are obtained by trial and
error
T1 = 2*0.2; T2 = 2*0.9; T3 = 2*0.7; T4 = 2*0.1; T5 =
```

```
4.15*0.2; T6 = 4.15*0.85;
Hrs = [zeros(1,9),T1,T2,
2*ones(1,8),T3,T4,0.00,zeros(1,6),-
0.02,T5,T6,4.15*ones(1,10)];
Hrs = [Hrs,fliplr(Hrs(2:end))];
k1 = 0:floor((M-1)/2); k2 = floor((M-1)/2)+1:M-1;
angH = [-alpha*(2*pi)/M*k1, alpha*(2*pi)/M*(M-k2)];
H = Hrs.*exp(j*angH);
h = real(ifft(H,M));
[Hr,w,a,L] = Hr_Type1(h);
%
% Computation of Exact band edges
w1 = 0;
w2 = w(max(find(Hr(I2:I3) <= 0.05)) + I2-1)/pi;
w3 = w(min(find(Hr(I2:I3) >= 1.90)) + I2-1)/pi;
w4 = w(max(find(Hr(I4:I5) >= 1.90)) + I4-1)/pi;
w5 = w(min(find(Hr(I4:I5+1) <= 0.05)) + I4-1)/pi;
w6 = w(max(find(Hr(I6:I7) <= 0.05)) + I6-1)/pi;
w7 = w(min(find(Hr(I6:I7) >= 4.00)) + I6-1)/pi;
w8 = 1;
%
% Computation of Exact tolerances
m1 = abs(min(Hr(I1:I2)));
m2 = max(Hr(I3:I4))-m(3);
m3 = abs(min(Hr(I5:I6)));
m4 = max(Hr(I7:I8))-m(7);
%
% Plot of Amplitude Response
Hf_3 = figure('paperunits','inches');
set(Hf_3,'NumberTitle','off','Name','P7.36c');
plot(w/pi,Hr,'g',wl(1:41)/pi,Hrs(1:41),'ro','linewidth',1
.5);
axis([0,1,-0.05,4.3]);
title('Amplitude Response in Part 3','fontsize',10);
xlabel('\omega/\pi','fontsize',10);
ylabel('Hr(w)','fontsize',10);grid;
set(gca,'XTickMode','manual','XTick',f);
set(gca,'YTickMode','manual','YTick',[-
0.05;0.05;1.9;2.1;4.0;4.3]);
%
% Printout of Order, Exact bandedges, and Exact
tolerances
disp(sprintf('\n(c) Frequency Sampling Design:'));
disp(sprintf(' Order : %2i',N));
```

```
disp(sprintf(' Exact band-
edges: %5.4f, %5.4f, %5.4f, %5.4f, %5.4f, %5.4f',...
w2, w3, w4, w5, w6, w7));
disp(sprintf(' Exact
tolerances: %5.4f, %5.4f, %5.4f, %5.4f,',...
m1, m2, m3, m4));
print -deps2 ../EPSFILES/P0736c
```

A plot of the amplitude response with grid-lines and axis labeling is shown in Figure 7.36.

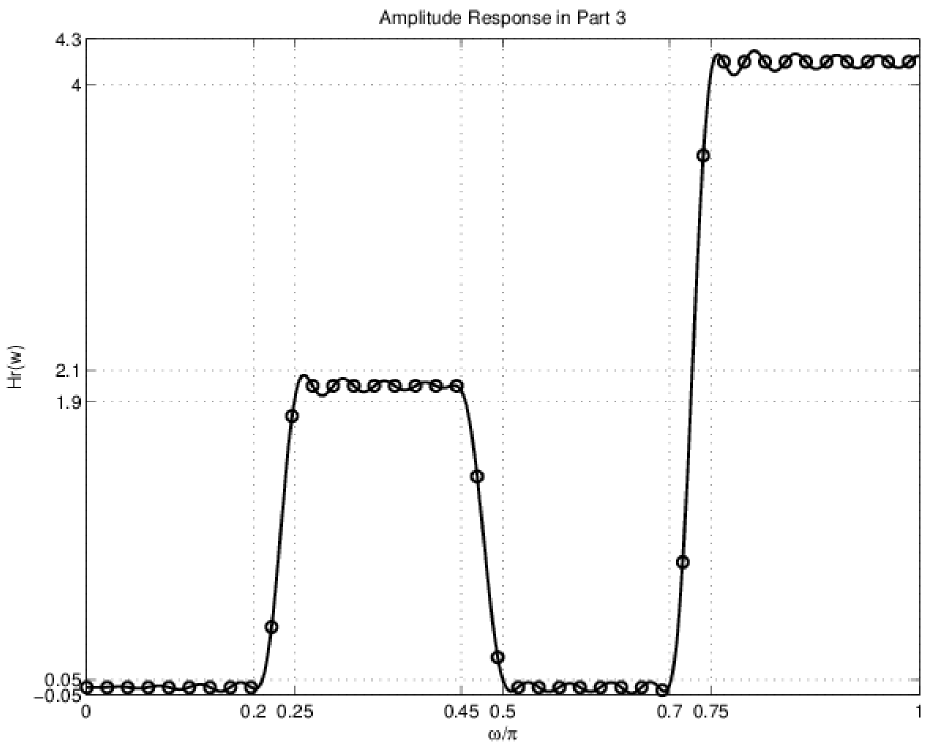


Figure 7.36: Amplitude Response Plot in Problem P7.36c

(d) Minimum-length linear-phase FIR filter design using the Parks-McClellan design approach to satisfy the given requirements and a plot of the amplitude response with grid-lines as shown in Figure P7.4:

```
%% P0736d.m
% (d) Equiripple design: The minimum M was found to be 65
%
weights = [d1/d1,d1/d2,d1/d3,d1/d4];
M = 65; N = M-1; h = firpm(M-1,f,m,weights);
[Hr,w,a,L] = Hr_Type1(h);
%
% Computation of Exact band edges
w1 = 0;
w2 = w(max(find(Hr(I2:I3) <= 0.05)) + I2-1)/pi;
w3 = w(min(find(Hr(I2:I3) >= 1.90)) + I2-1)/pi;
w4 = w(max(find(Hr(I4:I5) >= 1.90)) + I4-1)/pi;
```

```
w5 = w(min(find(Hr(I4:I5) <= 0.05)) + I4-1)/pi;
w6 = w(max(find(Hr(I6:I7) <= 0.05)) + I6-1)/pi;
w7 = w(min(find(Hr(I6:I7) >= 4.00)) + I6-1)/pi;
w8 = 1;
%
% Computation of Exact tolerances
m1 = abs(min(Hr(I1:I2)));
m2 = max(Hr(I3:I4))-m(3);
m3 = abs(min(Hr(I5:I6)));
m4 = max(Hr(I7:I8))-m(7);
%
% Plot of Amplitude Response
Hf_4 = figure('paperunits','inches');
set(Hf_4,'NumberTitle','off','Name','P7.36d');
plot(w/pi,Hr,'g','linewidth',1.5); axis([0,1,-0.05,4.3]);
title('Amplitude Response in Part 4','fontsize',10);
xlabel('\omega/\pi','fontsize',10);
ylabel('Hr(w)','fontsize',10);grid;
set(gca,'XTickMode','manual','XTick',f);
set(gca,'YTickMode','manual','YTick',[-
0.05;0.05;1.9;2.1;4.0;4.3]);
%
% Printout of Order, Exact bandedges, and Exact
tolerances
disp(sprintf('\n(d) Parks-McClellan Design:'));
disp(sprintf(' Order : %2i',N));
disp(sprintf(' Exact band-
edges: %5.4f, %5.4f, %5.4f, %5.4f, %5.4f, %5.4f',...
w2, w3, w4, w5, w6, w7));
disp(sprintf(' Exact
tolerances: %5.4f, %5.4f, %5.4f, %5.4f,',...
m1, m2, m3, m4));
print -deps2 ../EPSFILES/P0736d
```

A plot of the amplitude response with grid-lines and axis labeling is shown in Figure 7.37.

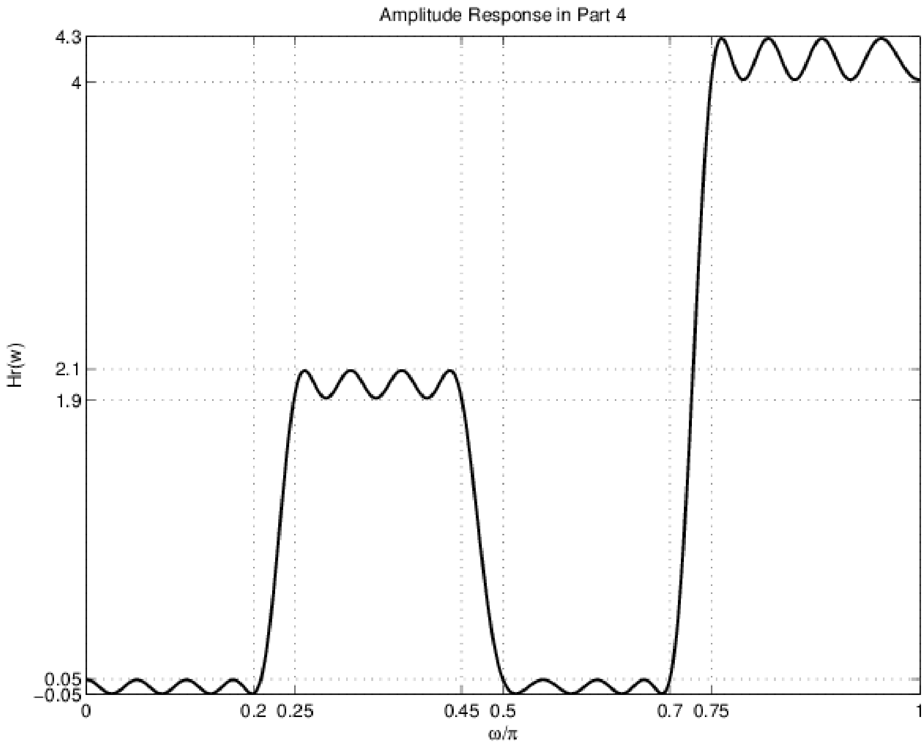


Figure 7.37: Amplitude Response Plot in Problem P7.36d

(e) Comparison of the above four design methods in terms of

• the *order* of the filter

• the *exact* band-edge frequencies

• the *exact* tolerances in each band

(a) Fixed Window Design:
Order : 118
Exact band-edges: 0.2020, 0.2460, 0.4540, 0.4980, 0.7000, 0.7480
Exact tolerances: 0.0127, 0.0127, 0.0262, 0.0263,

(b) Kaiser Window Design:
Order : 88
Exact band-edges: 0.2020, 0.2460, 0.4540, 0.5000, 0.7000, 0.7480
Exact tolerances: 0.0225, 0.0233, 0.0464, 0.0475,

(c) Frequency Sampling Design:
Order : 80
Exact band-edges: 0.2100, 0.2500, 0.4540, 0.5020, 0.7020, 0.7500
Exact tolerances: 0.0262, 0.0706, 0.0387, 0.0731,

(d) Parks-McClellan Design:
Order : 64
Exact band-edges: 0.2080, 0.2500, 0.4500, 0.5000, 0.7000, 0.7500
Exact tolerances: 0.0460, 0.0912, 0.0455, 0.1375

P7.37

Design a minimum-order linear-phase FIR filter, using the Parks-McClellan algorithm, to satisfy the requirements given in Figure P7.5. Provide a plot of the amplitude response with grid-lines as shown in Figure P7.5.

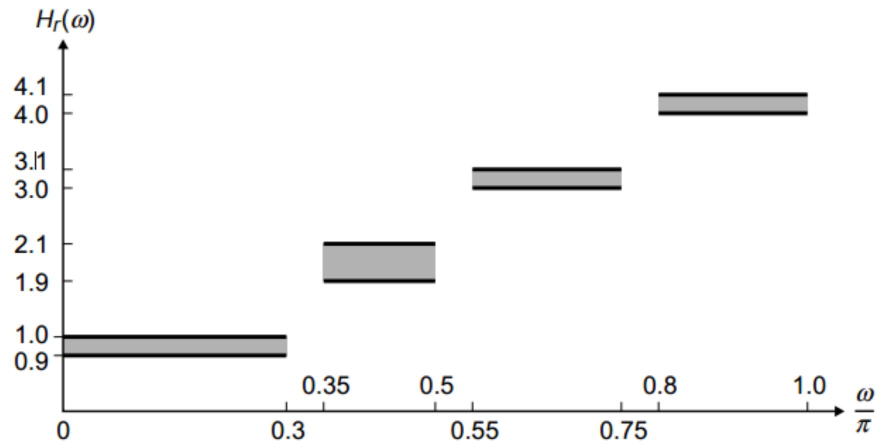


FIGURE P7.5 *Filter Specifications for Problem P7.37*

Solutions

Design of a minimum-order linear-phase FIR filter, using the Parks-McClellan algorithm, to satisfy the requirements given in Figure P7.5 and a plot of the amplitude response with grid-lines as shown in Figure P7.5:

```
% P7.37
clear;clc; close all;
% Specifications
f = [0,0.3,0.35,0.5,0.55,0.75,0.8,1];
m = [0.95,0.95,2,2,3.05,3.05,4.05,4.05];
d1 = 0.05; d2=0.1; d3=.05; d4=.05;
weights = [d4/d1,d4/d2,d4/d3,d4/d4];
% Optimum Design
h = firpm(50,f,m,weights); [Hr,w,a,L] = Hr_Type1(h);
Hr_min = min(Hr(401:501));
disp(sprintf('\n Achieved Redsponse in the Band-4
= %5.2f',Hr_min));
% Achieved Redsponse in the Band-4 = 4.01
%
Hf_1 = figure('Units','inches','position',[1,1,6,3],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.37');
plot(w/pi,Hr,'g','linewidth',1.5); axis([0,1,0,5]);
title('Amplitude Response','fontsize',10);
```

```
xlabel('\omega/\pi','fontsize',10);
ylabel('H_r(\omega)','fontsize',10)
set(gca,'XTickMode','manual','XTick',f); grid;
set(gca,'YTickMode','manual','YTick',[0.9,1.0,1.9,2.1,3.0
,3.1,4.0,4.1]);
print -deps2 ../EPSFILES/P0737
```

```
Achieved Redsponse in the Band-4 =  4.01
```

A plot of the amplitude response with grid-lines and axis labeling is shown in Figure 7.38.

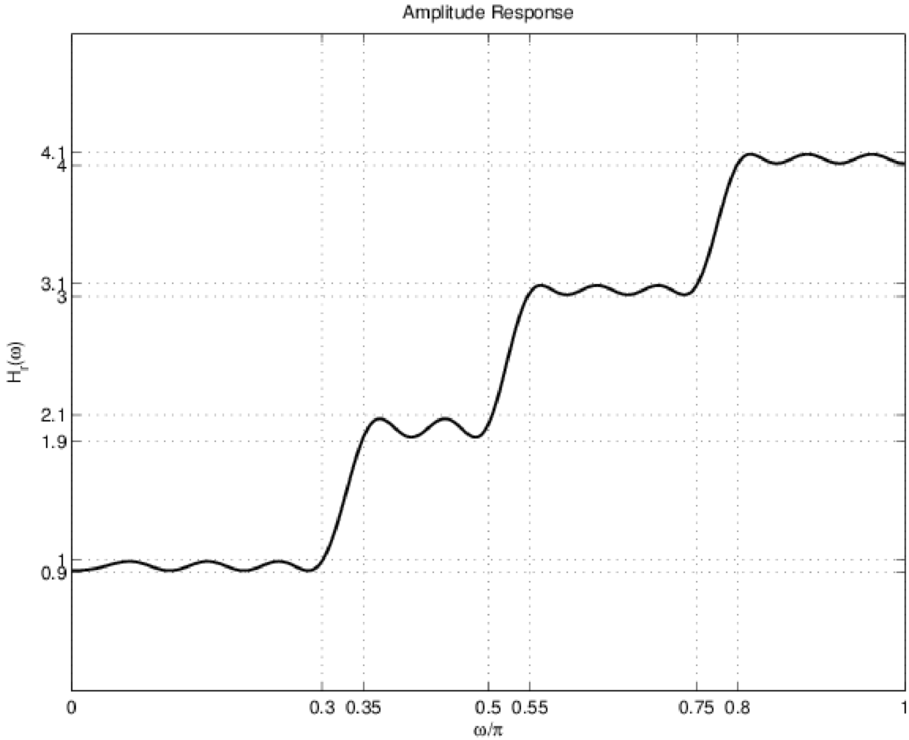


Figure 7.38: Amplitude Response Plot in Problem P7.37

P7.38

Design a minimum-length linear-phase bandpass filter of Problem P7.9 using the Parks-McClellan algorithm.

1. Plot the impulse response and the magnitude response in dB of the designed filter in one figure plot.
2. Plot the amplitude response of the designed filter and count the total number of extrema in passband and stopbands. Verify this number with the theoretical estimate of the total number of extrema.
3. Compare the order of this filter with that of the filter in Problem P7.9.

Solutions

Design of a minimum-length linear-phase bandpass filter of Problem P7.9 using the Parks-McClellan algorithm:

```
% P7.38
clear;clc; close all;
%% Specifications:
ws1 = 0.2*pi; % lower stopband edge
wp1 = 0.35*pi; % lower passband edge
wp2 = 0.55*pi; % upper passband edge
ws2 = 0.75*pi; % upper stopband edge
Rp = 0.25; % passband ripple
As = 40; % stopband attenuation
%
% Compute Tolerances
[delta1,delta2] = db2delta(Rp,As);
% Assemble Design Parameters
f = [0,ws1,wp1,wp2,ws2,pi]/pi; m = [0,0,1,1,0,0];
weights = [1,delta2/delta1,1];
% Optimum Design
M = 26; h = firpm(M-1,f,m,weights); n= 0:M-1;
% Response Plots
[db,mag,pha,grd,w] = freqz_m(h,1); delta_w = pi/500;
Rpd = -min(db((wp1/delta_w)+1:(wp2/delta_w)+1)), % Actual
passband ripple
Asd = floor(-max(db(1:(ws1/delta_w)+1))), % Actual Attn

Rpd =
    0.2170
Asd =
    41
```

(a) Plot of the impulse response and the magnitude response in dB of the designed filter:

```
%% P0738a.m
% 1. Filter Impulse and Log-Magnitude Response Plots
Hf_1 = figure('Units','inches','position',[1,1,6,4],...
'paperunits','inches');
set(Hf_1,'NumberTitle','off','Name','P7.38a');
subplot(2,1,1);
Hs_1 = stem(n,h,'g','filled'); set(Hs_1,'markersize',3);
title('Impulse Response','fontsize',10);
set(gca,'XTick',[0;M-1]); axis([-1,M,min(h)-
0.1,max(h)+0.1]);
```

```
xlabel('n','fontsize',10); ylabel('h(n)','fontsize',10)
subplot(2,1,2); plot(w/pi,db,'g','linewidth',1.5);
title('Magnitude Response in dB','fontsize',10);
axis([0,1,-As-30,5]);
xlabel('\omega/\pi','fontsize',10);
ylabel('Decibels','fontsize',10)
set(gca,'XTick',[0;0.2;0.35;0.55;0.75;1])
set(gca,'XTickLabel',{' 0 ';'0.2 ';'0.35';'0.55';'0.75';'
1 '},'fontsize',8)
set(gca,'YTick',[-40;0]); set(gca,'YTickLabel',[' 40';' 0
']);grid
print -deps2 ../EPSFILES/P0738a
```

The plots are shown in Figure 7.39.

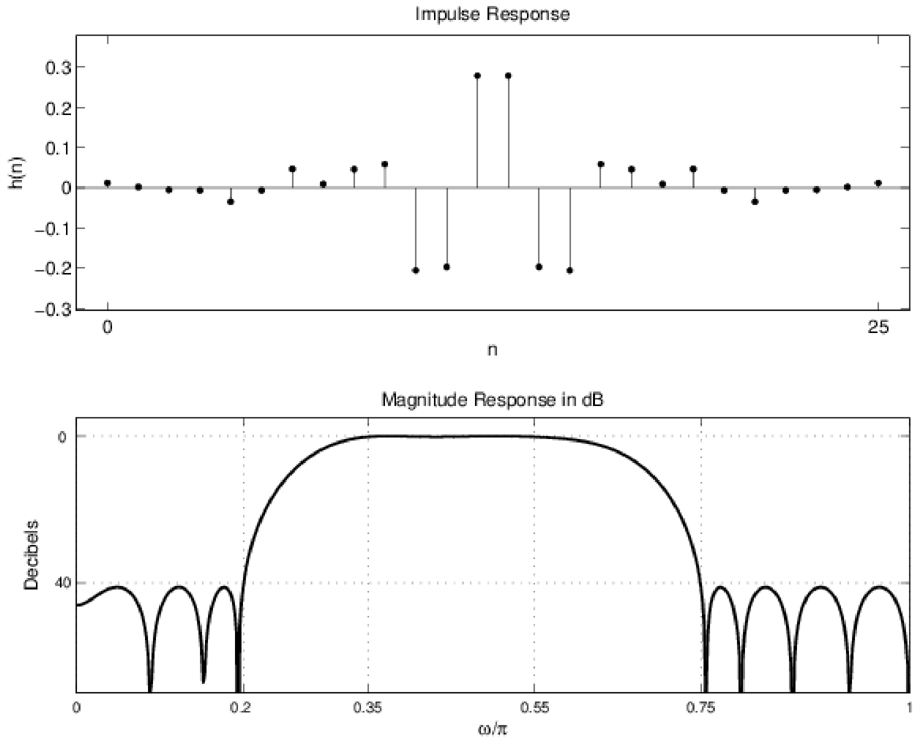


Figure 7.39: Amplitude Response Plot in Problem P7.38a

(b) Plot the amplitude response of the designed filter:

```
%% P0738b.m
% 2. Amplitude Response plot
[Hr,w,a,L] = Hr_Type2(h);
Hf_2 = figure('Units','inches','position',[1,1,6,3],...
'paperunits','inches');
set(Hf_2,'NumberTitle','off','Name','P7.38b');
plot(w/pi,Hr,'g','linewidth',1.5);
title('Amplitude Response','fontsize',10); axis([0,1,-
0.2,1.2]);
xlabel('\omega/\pi','fontsize',10);
```

```
ylabel('Amplitude','fontsize',10)
set(gca,'XTick',[0;0.2;0.35;0.55;0.75;1]);
set(gca,'XTickLabel',{' 0 ';'0.2 ';'0.35';'0.55';'0.75';'
1 '})
set(gca,'YTick',[-delta2;delta2;1-delta1;1+delta1]);
grid;
print -deps2 ../EPSFILES/P0738b
```

The plot is shown in Figure 7.40. From Figure 7.40, the total number of extrema in the error function in the passband and stopbands are 14. Since M = 26 for this design, L = M/2 - 1 = 12. Then the extrema are L + 2 or L + 3. Hence this is a L + 2 = 14 equiripple design.

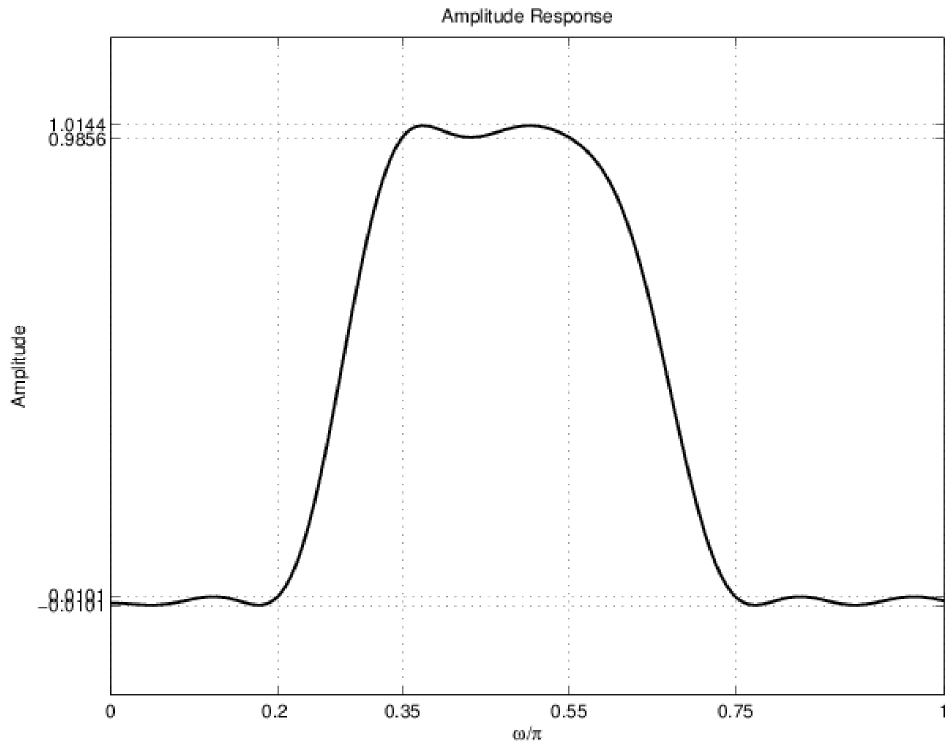


Figure 7.40: Amplitude Response Plot in Problem P7.38b

(c) The order of this filter is M-1 = 25 while that of the filter designed in Problem P7.9 is M-1 = 43-1 = 42. Thus this is an optimum design.

Chapter 8

EXAMPLE 8.1

Given that $\left|H_a(j\Omega)\right|^2 = \dfrac{1}{1+64\Omega^6}$, determine the analog filter system function $H_a(s)$.

Solutions

From the given magnitude-squared response,

$$|H_a(j\Omega)|^2 = \frac{1}{1+64\Omega^6} = \frac{1}{1+\left(\frac{\Omega}{0.5}\right)^{2(3)}}$$

Comparing this with expression (8.45), we obtain $N = 3$ and $\Omega c = 0.5$. The poles of $Ha(s)Ha(-s)$ are as shown in Figure 8.14.
Hence

$$\begin{aligned} H_a(j\Omega) &= \frac{\Omega_c^3}{(s-p_2)(s-p_3)(s-p_4)} \\ &= \frac{1/8}{(s+0.25-j0.433)(s+0.5)(s+0.25+j0.433)} \\ &= \frac{0.125}{(s+0.5)(s^2+0.5s+0.25)} \end{aligned}$$

EXAMPLE 8.2

Design a 3rd-order Butterworth analog prototype filter with Ω_c = 0.5 given in Example 8.1.

Solutions

```
%% Example 8.2
clear;clc;close all;
N = 3; OmegaC = 0.5; [b,a] = u_buttap(N,OmegaC);
[C,B,A] = sdir2cas(b,a)

C =
    0.1250
B =
     0     0     1
A =
```

```
    1.0000    0.5000    0.2500
         0    1.0000    0.5000
```

The cascade form coefficients agree with those in Example 8.1.

EXAMPLE 8.3

Design a lowpass Butterworth filter to satisfy
Passband cutoff: $\Omega_p = 0.2\pi$; Passband ripple: $R_p = 7\text{dB}$
Stopband cutoff: $\Omega_s = 0.3\pi$; Stopband ripple: $A_s = 16\text{dB}$

Solutions

From (8.49)

$$N = \left\lceil \frac{\log_{10}\left[\left(10^{0.7}-1\right)/\left(10^{1.6}-1\right)\right]}{2\log_{10}\left(0.2\pi/0.3\pi\right)} \right\rceil = \lceil 2.79 \rceil = 3$$

To satisfy the specifications exactly at Ωp, from (8.50) we obtain

$$\Omega_c = \frac{0.2\pi}{\sqrt[6]{(10^{0.7}-1)}} = 0.4985$$

To satisfy specifications exactly at Ωs, from (8.51) we obtain

$$\Omega_c = \frac{0.3\pi}{\sqrt[6]{(10^{1.6}-1)}} = 0.5122$$

Now we can choose any Ωc between the above two numbers. Let us choose $\Omega c = 0.5$. We have to design a Butterworth filter with $N = 3$ and $\Omega c = 0.5$, which we did in Example 8.1. Hence

$$H_a(j\Omega) = \frac{0.125}{(s+0.5)\,(s^2+0.5s+0.25)}$$

EXAMPLE 8.4

Design the analog Butterworth lowpass filter specified in Example 8.3 using MATLAB.

Solutions

MATLAB script:

```
%% Example 8.4
clear;clc;close all;
Wp = 0.2*pi; Ws = 0.3*pi; Rp = 7; As = 16;
Ripple = 10^(-Rp/20); Attn  =10^(-As/20);
% Analog filter design:
[b,a] = afd_butt(Wp,Ws,Rp,As);
```

```
% Calculation of second-order sections:
[C,B,A] = sdir2cas(b,a)
% Calculation of Frequency Response:
[db,mag,pha,w] = freqs_m(b,a,0.5*pi);
% Calculation of Impulse response:
[ha,x,t] = impulse(b,a);

*** Butterworth Filter Order =  3
C =
   0.1238
B =
    0     0     1
A =
   1.0000    0.4985    0.2485
        0    1.0000    0.4985
```

The system function is given by

$$H_a(s) = \frac{0.1238}{(s^2 + 0.4985s + 0.2485)\,(s + 0.4985)}$$

This $H_a(s)$ is slightly different from the one in Example 8.3 because in that example we used $\Omega_c = 0.5$, while in the **afd_butt** function Ω_c is chosen to satisfy the specifications at Ω_p. The filter plots are shown in Figure 8.15.

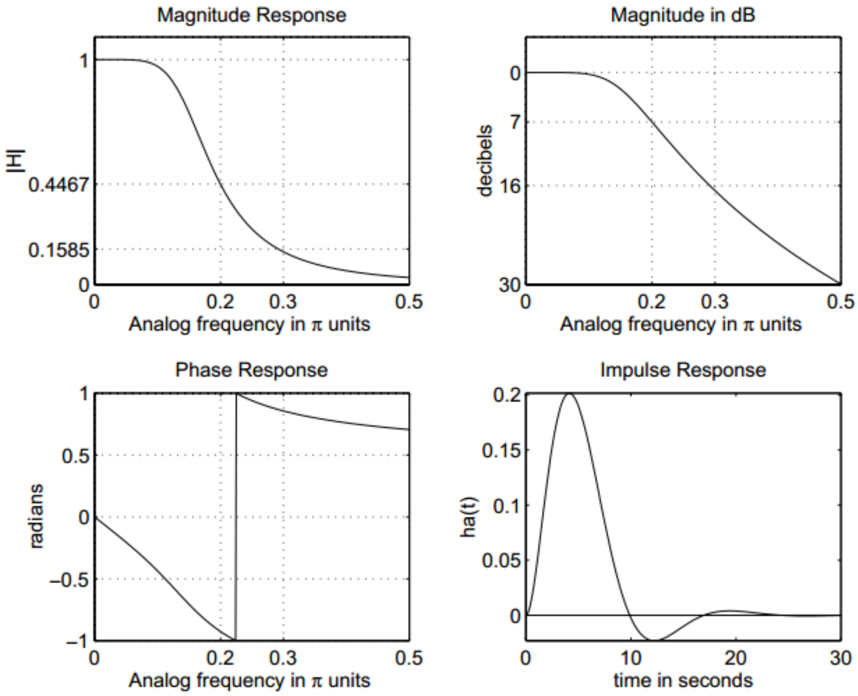


FIGURE 8.15 *Butterworth analog filter in Example 8.4*

EXAMPLE 8.5

Design a lowpass Chebyshev-I filter to satisfy

Passband cutoff: $\Omega_p = 0.2\pi$; Passband ripple: $R_p = 1\text{dB}$

Stopband cutoff: $\Omega_s = 0.3\pi$; Stopband ripple: $A_s = 16$dB

Solutions

First compute the necessary parameters.

$$\epsilon = \sqrt{10^{0.1(1)} - 1} = 0.5088 \qquad A = 10^{16/20} = 6.3096$$

$$\Omega_c = \Omega_p = 0.2\pi \qquad \Omega_r = \frac{0.3\pi}{0.2\pi} = 1.5$$

$$g = \sqrt{(A^2 - 1)/\epsilon^2} = 12.2429 \qquad N = 4$$

Now we can determine $H_a(s)$.

$$\alpha = \frac{1}{\epsilon} + \sqrt{1 + \frac{1}{\epsilon^2}} = 4.1702$$

$$a = 0.5\left(\sqrt[N]{\alpha} - \sqrt[N]{1/\alpha}\right) = 0.3646$$

$$b = 0.5\left(\sqrt[N]{\alpha} + \sqrt[N]{1/\alpha}\right) = 1.0644$$

There are four poles for $H_a(s)$:

$$p_{0,3} = (a\Omega_c)\cos\left[\frac{\pi}{2} + \frac{\pi}{8}\right] \pm (b\Omega_c)\sin\left[\frac{\pi}{2} + \frac{\pi}{8}\right] = -0.0877 \pm j0.6179$$

$$p_{1,2} = (a\Omega_c)\cos\left[\frac{\pi}{2} + \frac{3\pi}{8}\right] \pm (b\Omega_c)\sin\left[\frac{\pi}{2} + \frac{3\pi}{8}\right] = -0.2117 \pm j0.2559$$

Hence

$$H_a(s) = \frac{K}{\prod_{k=0}^{3}(s - p_k)} = \frac{\overbrace{0.89125 \times .1103 \times .3895}^{0.03829}}{(s^2 + 0.1754s + 0.3895)(s^2 + 0.4234s + 0.1103)}$$

Note that the numerator is such that

$$H_a(j0) = \frac{1}{\sqrt{1 + \epsilon^2}} = 0.89125$$

EXAMPLE 8.6

Design the analog Chebyshev-I lowpass filter given in Example 8.5 using MATLAB.

Solutions

MATLAB script:

```
%% Example 8.6
clear;clc;close all;
Wp = 0.2*pi; Ws = 0.3*pi; Rp = 1; As = 16;
Ripple = 10 ^ (-Rp/20); Attn = 10 ^ (-As/20);
```

```
% Analog filter design:
[b,a] = afd_chb1(Wp,Ws,Rp,As);
% Calculation of second-order sections:
[C,B,A] = sdir2cas(b,a)
% Calculation of Frequency Response:
[db,mag,pha,w] = freqs_m(b,a,0.5*pi);
% Calculation of Impulse response:
[ha,x,t] = impulse(b,a);

*** Chebyshev-1 Filter Order =  4
C =
    0.0383
B =
     0     0     1
A =
    1.0000    0.4233    0.1103
    1.0000    0.1753    0.3895
```

The specifications are satisfied by a 4th-order Chebyshev-I filter whose system function is

$$H_a(s) = \frac{0.0383}{(s^2 + 4233s + 0.1103)\,(s^2 + 0.1753s + 0.3895)}$$

The filter plots are shown in Figure 8.16.

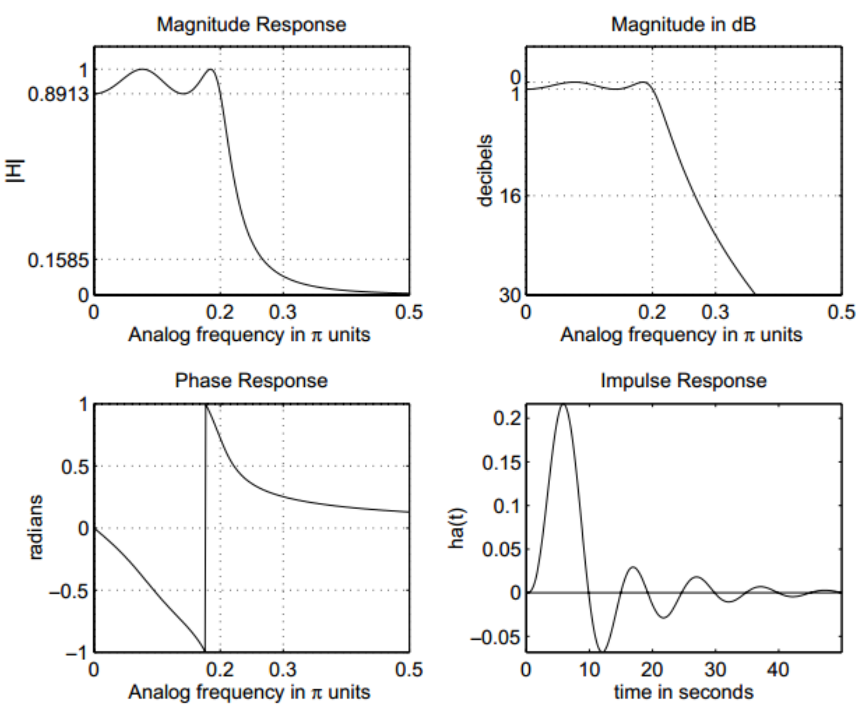


FIGURE 8.16 *Chebyshev-I analog filter in Example 8.6*

EXAMPLE 8.7

Design a Chebyshev-II analog lowpass filter to satisfy the specifications given in Example 8.5:

Passband cutoff: $\Omega_p = 0.2\pi$; Passband ripple: $R_p = 1$dB

Stopband cutoff: $\Omega_s = 0.3\pi$; Stopband ripple: $A_s = 16$dB

Solutions

MATLAB script:

```
%% Example 8.7
clear;clc;close all;
Wp = 0.2*pi; Ws = 0.3*pi; Rp = 1; As = 16;
Ripple = 10 ^ (-Rp/20); Attn = 10 ^ (-As/20);
% Analog filter design:
[b,a] = afd_chb2(Wp,Ws,Rp,As);
% Calculation of second-order sections:
[C,B,A] = sdir2cas(b,a)
% Calculation of Frequency Response:
[db,mag,pha,w] = freqs_m(b,a,0.5*pi);
% Calculation of Impulse response:
[ha,x,t] = impulse(b,a);

*** Chebyshev-2 Filter Order =  4
C =
    0.1585
B =
    1.0000    0.0000    6.0654
    1.0000         0    1.0407
A =
    1.0000    1.9521    1.4747
    1.0000    0.3719    0.6784
```

The specifications are satisfied by a 4th-order Chebyshev-II filter whose system function is

$$H_a(s) = \frac{0.1585\left(s^2+6.0654\right)\left(s^2+1.0407\right)}{\left(s^2+1.9521s+1.4747\right)\left(s^2+0.3719s+0.6784\right)}$$

The filter plots are shown in Figure 8.17.

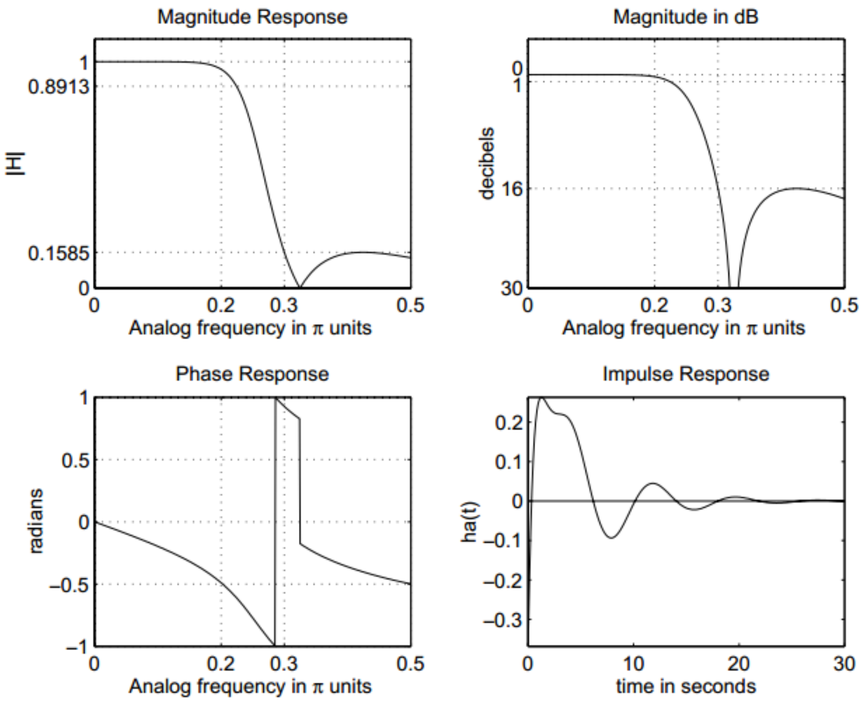


FIGURE 8.17 *Chebyshev-II analog filter in Example 8.7*

EXAMPLE 8.8

Design an analog elliptic lowpass filter to satisfy the following specifications of Example 8.5:

$$\Omega_p = 0.2\pi, \ R_p = 1 \text{ dB}$$
$$\Omega_s = 0.3\pi, \ A_s = 16 \text{ db}$$

Solution

MATLAB script:

```
%% Example 8.8
clear;clc;close all;
Wp = 0.2*pi; Ws = 0.3*pi; Rp = 1; As = 16;
Ripple = 10 ^ (-Rp/20); Attn = 10 ^ (-As/20);
% Analog filter design:
[b,a] = afd_elip(Wp,Ws,Rp,As);
% Calculation of second-order sections:
[C,B,A] = sdir2cas(b,a)
% Calculation of Frequency Response:
[db,mag,pha,w] = freqs_m(b,a,0.5*pi);
% Calculation of Impulse response:
[ha,x,t] = impulse(b,a);

*** Elliptic Filter Order =  3
```

```
C =
    0.2740
B =
    1.0000         0    0.6641
A =
    1.0000    0.1696    0.4102
         0    1.0000    0.4435
```

The specifications are satisfied by a 3rd-order elliptic filter whose system function is

$$H_a(s) = \frac{0.274\left(s^2 + 0.6641\right)}{\left(s^2 + 0.1696s + 0.4102\right)\left(s + 0.4435\right)}$$

The filter plots are shown in Figure 8.18.

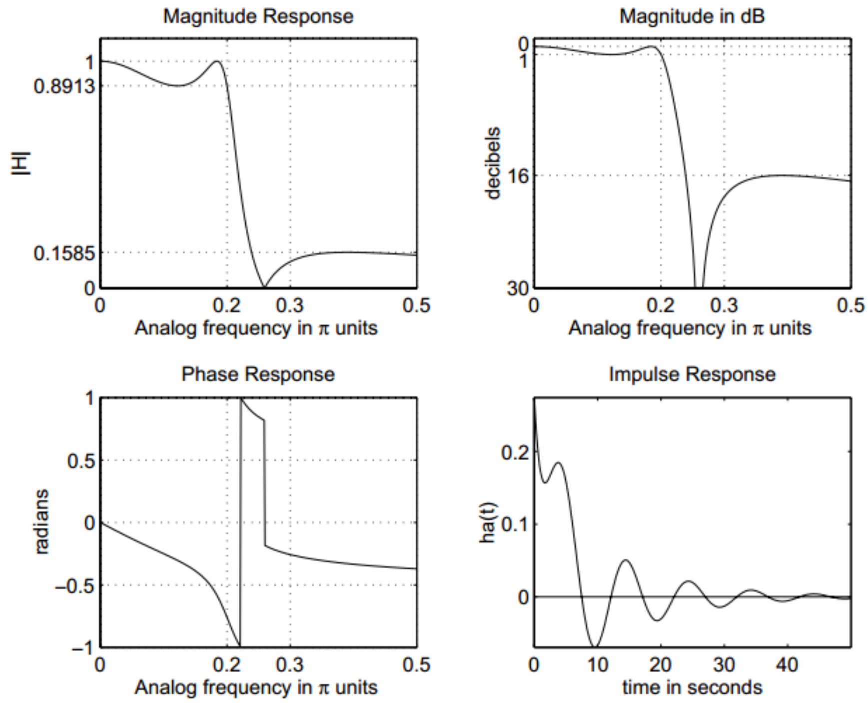


FIGURE 8.18 *Elliptic analog lowpass filter in Example 8.8*

EXAMPLE 8.9

Transform

$$H_a(s) = \frac{s+1}{s^2+5s+6}$$

into a digital filter $H(z)$ using the impulse invariance technique in which $T = 0.1$.

Solutions

We first expand $H_a(s)$ using partial fraction expansion:

$$H_a(s) = \frac{s+1}{s^2+5s+6} = \frac{2}{s+3} - \frac{1}{s+2}$$

The poles are at p_1 = -3 and p_2 = -2. Then from (8.64) and using T = 0.1, we obtain

$$H(z) = \frac{2}{1-e^{-3T}z^{-1}} - \frac{1}{1-e^{-2T}z^{-1}} = \frac{1-0.8966z^{-1}}{1-1.5595z^{-1}+0.6065z^{-2}}$$

It is easy to develop a MATLAB function to implement the impulse invariance mapping. Given a rational function description of $H_a(s)$, we can use the **residuez** function to obtain its pole-zero description. Then each analog pole is mapped into a digital pole using (8.63). Finally, the residuez function can be used to convert $H(z)$ into rational function form. This procedure is given in the function **imp_invr**.

```
function [b,a] = imp_invr(c,d,T)
% Impulse Invariance Transformation from Analog to
Digital Filter
% ---------------------------------------------------------
---------
% [b,a] = imp_invr(c,d,T)
% b = Numerator polynomial in z^(-1) of the digital
filter
% a = Denominator polynomial in z^(-1) of the digital
filter
% c = Numerator polynomial in s of the analog filter
% d = Denominator polynomial in s of the analog filter
% T = Sampling (transformation) parameter
%
[R,p,k] = residue(c,d); p = exp(p*T);
[b,a] = residuez(R,p,k); b = real(b'); a = real(a');
```

A similar function called **impinvar** is available in the SP toolbox of MATLAB.

EXAMPLE 8.10

We demonstrate the use of the **imp_invr** function on the system function from Example 8.9

Solutions

MATLAB script:

```
%% Example 8.10
clear;clc;close all;
c = [1,1]; d = [1,5,6]; T = 0.1;
[b,a] = imp_invr(c,d,T)

b =
    1.0000
```

```
   -0.8966
a =
    1.0000
   -1.5595
    0.6065
```

The digital filter is

$$H(z) = \frac{1 - 0.8966z^{-1}}{1 - 1.5595z^{-1} + 0.6065z^{-2}}$$

as expected. In Figure 8.20 we show the impulse responses and the magnitude responses (plotted up to the sampling frequency $1/T$) of the analog and the resulting digital filter. Clearly, the aliasing in the frequency domain is evident.

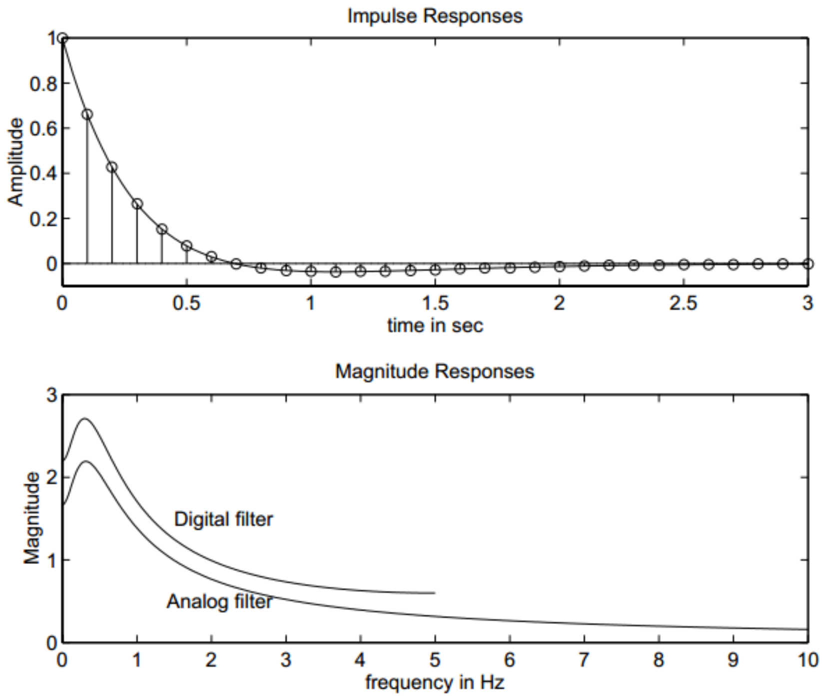


FIGURE 8.20 *Impulse and frequency response plots in Example 8.10*

EXAMPLE 8.11

Design a lowpass digital filter using a Butterworth prototype to satisfy

$$\omega_p = 0.2\pi, \ R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi, \ A_s = 15 \text{ dB}$$

Solutions

The design procedure is described in the following MATLAB script:

```
%% Example 8.11
clear;clc;close all;
% Digital Filter Specifications:
```

```
wp = 0.2*pi; % digital Passband freq in Hz
ws = 0.3*pi; % digital Stopband freq in Hz
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
% Analog Prototype Specifications: Inverse mapping for
frequencies
T = 1; % Set T=1
OmegaP = wp / T; % Prototype Passband freq
OmegaS = ws / T; % Prototype Stopband freq
% Analog Butterworth Prototype Filter Calculation:
[cs,ds] = afd_butt(OmegaP,OmegaS,Rp,As);
% Impulse Invariance transformation:
[b,a] = imp_invr(cs,ds,T); [C,B,A] = dir2par(b,a)

*** Butterworth Filter Order =  6
C =
     []
B =
    1.8557   -0.6304
   -2.1428    1.1454
    0.2871   -0.4466
A =
    1.0000   -0.9973    0.2570
    1.0000   -1.0691    0.3699
    1.0000   -1.2972    0.6949
```

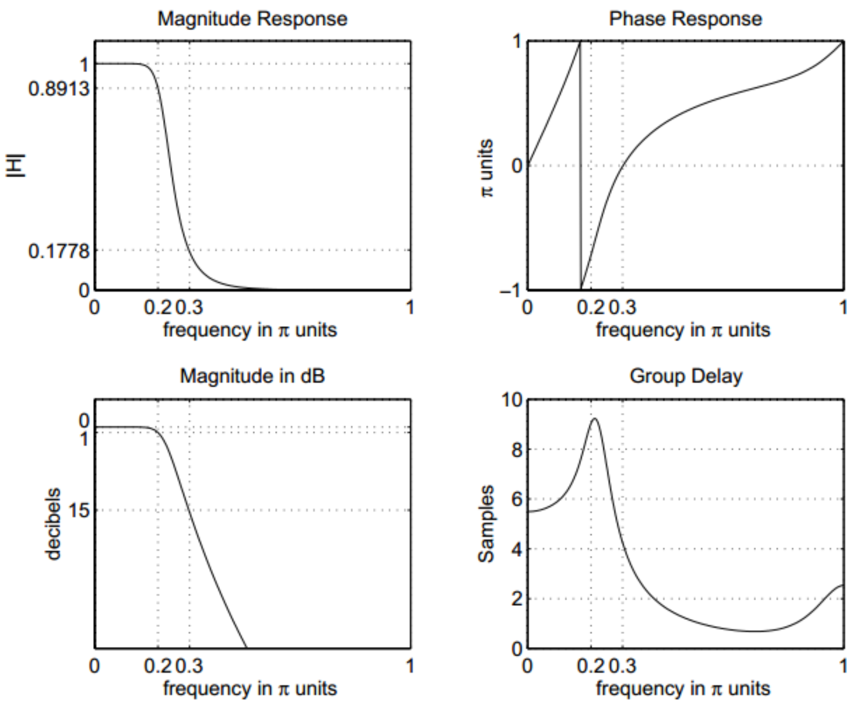


FIGURE 8.21 *Digital Butterworth lowpass filter using impulse invariance design*

The desired filter is a 6th-order Butterworth filter whose system function $H(z)$ is given in the parallel form

$$H(z) = \frac{1.8587 - 0.6304z^{-1}}{1 - 0.9973z^{-1} + 0.257z^{-2}} + \frac{-2.1428 + 1.1454z^{-1}}{1 - 1.0691z^{-1} + 0.3699z^{-2}} + \frac{0.2871 - 0.4463z^{-1}}{1 - 1.2972z^{-1} + 0.6449z^{-2}}$$

The frequency response plots are given in Figure 8.21.

EXAMPLE 8.12

Design a lowpass digital filter using a Chebyshev-I prototype to satisfy

$$\omega_p = 0.2\pi,\ R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi,\ A_s = 15 \text{ dB}$$

Solutions

```
%% Example 8.12
clear;clc;close all;
wp = 0.2*pi; % digital Passband freq in Hz
ws = 0.3*pi; % digital Stopband freq in Hz
Rp = 1; % Passband ripple in dB
As = 15;% Stopband attenuation in dB
% % Analog Prototype Specifications: Inverse mapping for
frequencies
T = 1;% Set T=1
OmegaP = wp/T;% Prototype Passband freq
OmegaS = ws/T;% Prototype Stopband freq
% % Analog Butterworth Prototype Filter Calculation:
[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As);
% % Impulse Invariance transformation:
[b,a] = imp_invr(cs,ds,T); [C,B,A] = dir2par(b,a)

*** Chebyshev-1 Filter Order =  4
C =
     []
B =
   -0.0833   -0.0246
    0.0833    0.0239
A =
    1.0000   -1.4934    0.8392
    1.0000   -1.5658    0.6549
```

The desired filter is a 4th-order Chebyshev-I filter whose system function $H(z)$ is

$$H(z) = \frac{-0.0833 - 0.0246z^{-1}}{1 - 1.4934z^{-1} + 0.8392z^{-2}} + \frac{-0.0833 + 0.0239z^{-1}}{1 - 1.5658z^{-1} + 0.6549z^{-2}}$$

The frequency response plots are given in Figure 8.22.

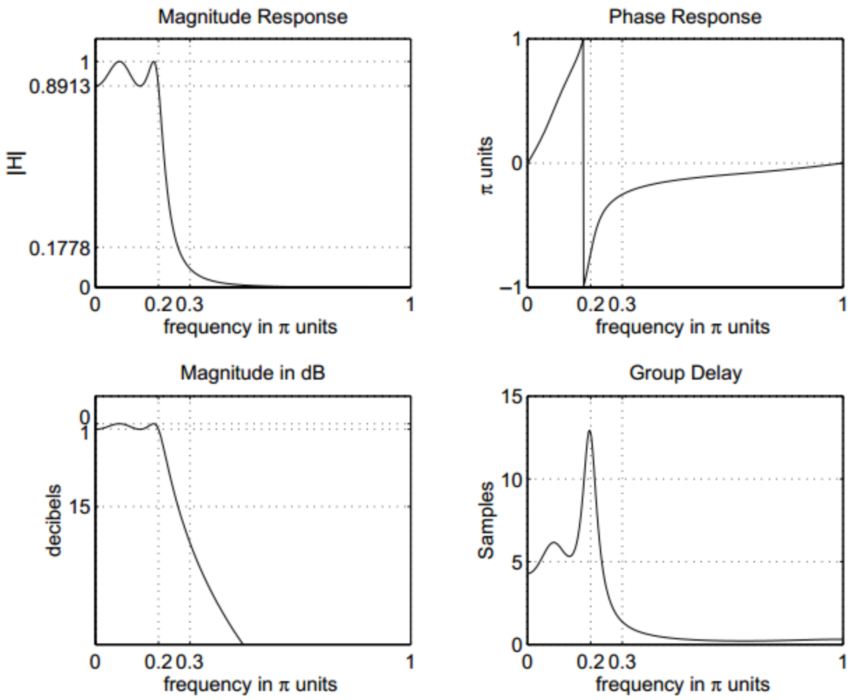


FIGURE 8.22 *Digital Chebyshev-I lowpass filter using impulse invariance design*

EXAMPLE 8.13

Design a lowpass digital filter using a Chebyshev-II prototype to satisfy

$$\omega_p = 0.2\pi, \quad R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi, \quad A_s = 15 \text{ dB}$$

Solutions

Recall that the Chebyshev-II filter is equiripple in the stopband. It means that this analog filter has a response that does not go to zero at high frequencies in the stopband. Therefore after impulse invariance transformation, the aliasing effect will be significant; this can degrade the passband response. The MATLAB script follows:

```
%% Example 8.13
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; % digital Passband freq in rad
ws = 0.3*pi; % digital Stopband freq in rad
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
```

```
% Analog Prototype Specifications: Inverse mapping for
frequencies
T = 1; % Set T=1
OmegaP = wp / T; % Prototype Passband freq
OmegaS = ws / T; % Prototype Stopband freq
% Analog Chebyshev-1 Prototype Filter Calculation:
[cs,ds] = afd_chb2(OmegaP,OmegaS,Rp,As);
% Impulse Invariance transformation:
[b,a] = imp_invr(cs,ds,T); [C,B,A] = dir2par(b,a);

*** Chebyshev-2 Filter Order =  4
```

From the frequency response plots in Figure 8.23 we clearly observe the passband as well as stopband degradation. Hence the impulse invariance design technique has failed to produce a desired digital filter.

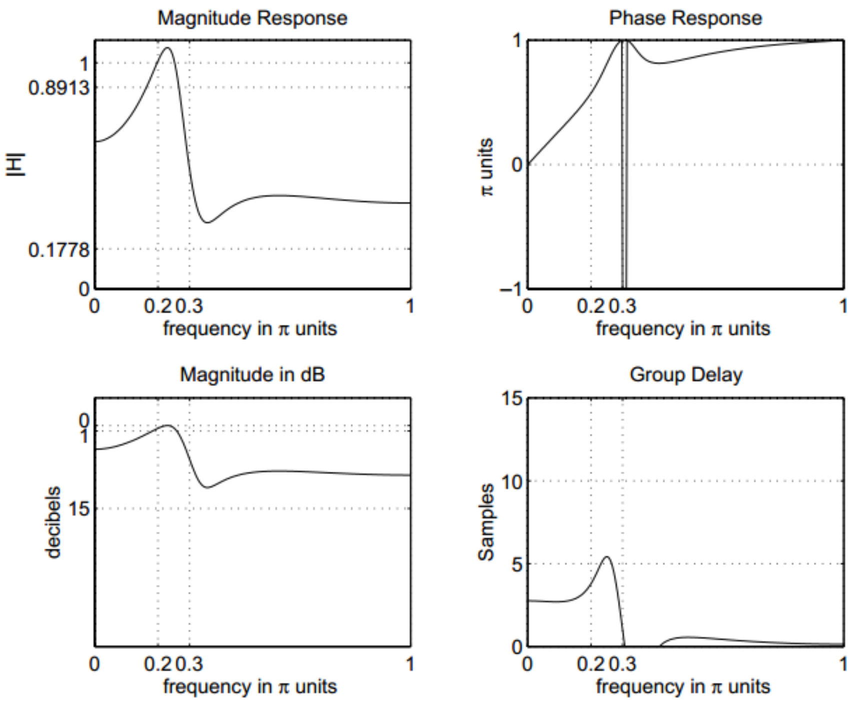


FIGURE 8.23 *Digital Chebyshev-II lowpass filter using impulse invariance design*

EXAMPLE 8.14

Design a lowpass digital filter using an elliptic prototype to satisfy

$$\omega_p = 0.2\pi, \ R_p = 1 \text{ dB}$$

$$\omega_s = 0.3\pi, \ A_s = 15 \text{ dB}$$

The elliptic filter is equiripple in both bands. Hence this situation is similar to that of the Chebyshev-II filter, and we should not expect a good digital filter. The MATLAB script follows:

```
%% Example 8.14
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; % digital Passband freq in rad
ws = 0.3*pi; % digital Stopband freq in rad
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
% Analog Prototype Specifications: Inverse mapping for
frequencies
T = 1; % Set T=1
OmegaP = wp / T; % Prototype Passband freq
OmegaS = ws / T; % Prototype Stopband freq
% Analog Elliptic Prototype Filter Calculation:
[cs,ds] = afd_elip(OmegaP,OmegaS,Rp,As);
% Impulse Invariance transformation:
[b,a] = imp_invr(cs,ds,T); [C,B,A] = dir2par(b,a);

*** Elliptic Filter Order =  3
```

From the frequency response plots in Figure 8.24 we clearly observe that once again the impulse invariance design technique has failed.

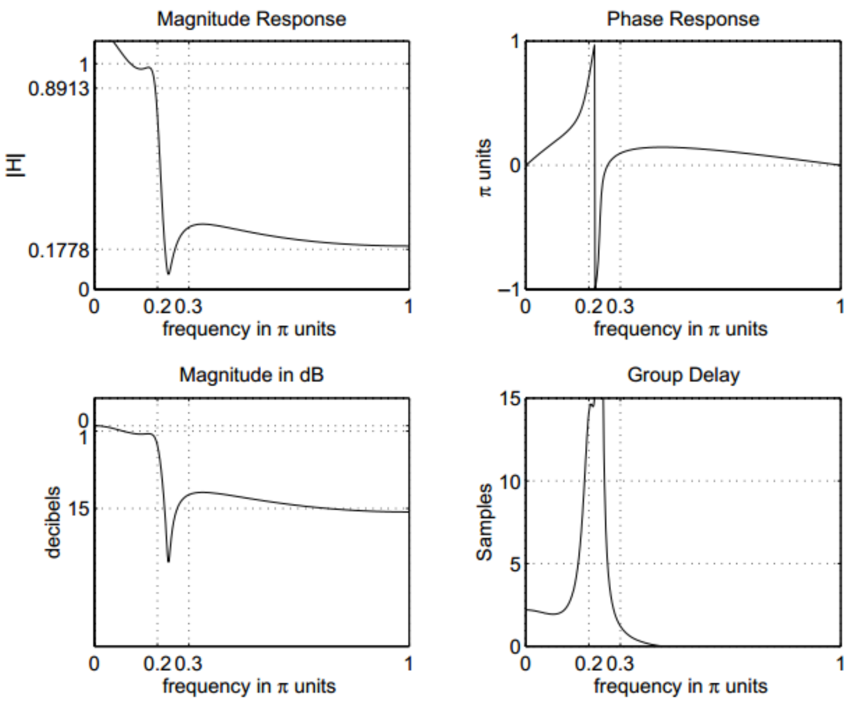


FIGURE 8.24 *Digital elliptic lowpass filter using impulse invariance design*

EXAMPLE 8.15

Transform $H_a(s) = \dfrac{s+1}{s^2+5s+6}$ into a digital filter using the bilinear transformation. Choose $T = 1$.

Solutions

Using (8.65), we obtain

$$H(z) = H_a\left(\frac{2}{T}\bigg|\left.\frac{1-z^{-1}}{1+z^{-1}}\right|_{T=1}\right) = H_a\left(2\frac{1-z^{-1}}{1+z^{-1}}\right)$$
$$= \frac{2\dfrac{1-z^{-1}}{1+z^{-1}}+1}{\left(2\dfrac{1-z^{-1}}{1+z^{-1}}\right)^2+5\left(2\dfrac{1-z^{-1}}{1+z^{-1}}\right)+6}$$

Simplifying,

$$H(z) = \frac{3+2z^{-1}-z^{-2}}{20+4z^{-1}} = \frac{0.15+0.1z^{-1}-0.05z^{-2}}{1+0.2z^{-1}}$$

MATLAB provides a function called bilinear to implement this mapping. Its invocation is similar to the **imp_invr** function, but it also takes several forms for different input-output quantities. The SP toolbox manual should be consulted for more details. Its use is shown in the following example.

EXAMPLE 8.16

Transform the system function *Ha*(*s*) in Example 8.15 using the **bilinear** function.

Solutions

MATLAB script:

```
%% Example 8.16
clear;clc;close all;
 c = [1,1]; d = [1,5,6]; T = 1; Fs = 1/T;
[b,a] = bilinear(c,d,Fs)

b =
    0.1500    0.1000   -0.0500
a =
```

```
    1.0000    0.2000   -0.0000
```

The filter is

$$H(z) = \frac{0.15 + 0.1z^{-1} - 0.05z^{-2}}{1 + 0.2z^{-1}}$$

as before.

EXAMPLE 8.17

Design the digital Butterworth filter of Example 8.11. The specifications are

$$\omega_p = 0.2\pi, \ R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi, \ A_s = 15 \text{ dB}$$

Solutions

MATLAB script:

```
%% Example 8.17
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; % digital Passband freq in rad
ws = 0.3*pi; % digital Stopband freq in rad
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
% Analog Prototype Specifications: Inverse mapping for
frequencies
T = 1; Fs = 1/T; % Set T=1
OmegaP = (2/T)*tan(wp/2); % Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(ws/2); % Prewarp Prototype Stopband
freq
% Analog Butterworth Prototype Filter Calculation:
[cs,ds] = afd_butt(OmegaP,OmegaS,Rp,As);
% Bilinear transformation:
[b,a] = bilinear(cs,ds,Fs); [C,B,A] = dir2cas(b,a)

*** Butterworth Filter Order =  6
C =
   5.7969e-04
B =
    1.0000    2.0320    1.0323
    1.0000    1.9997    1.0000
    1.0000    1.9683    0.9687
A =
```

```
    1.0000   -0.9459    0.2342
    1.0000   -1.0541    0.3753
```

The desired filter is once again a 6th-order filter and has 6 zeros. Since the 6th-order zero of $H_a(s)$ at s = -∞ is mapped to z = -1, these zeros should be at z = -1. Due to the finite precision of MATLAB these zeros are not exactly at z = -1. Hence the system function should be

$$H(z) = \frac{0.00057969\left(1+z^{-1}\right)^6}{(1-0.9459z^{-1}+0.2342z^{-2})\,(1-1.0541z^{-1}+0.3753z^{-2})\,(1-1.3143z^{-1}+0.7149z^{-2})}$$

The frequency response plots are given in Figure 8.26. Comparing these plots with those in Figure 8.21, we observe that these two designs are very similar.

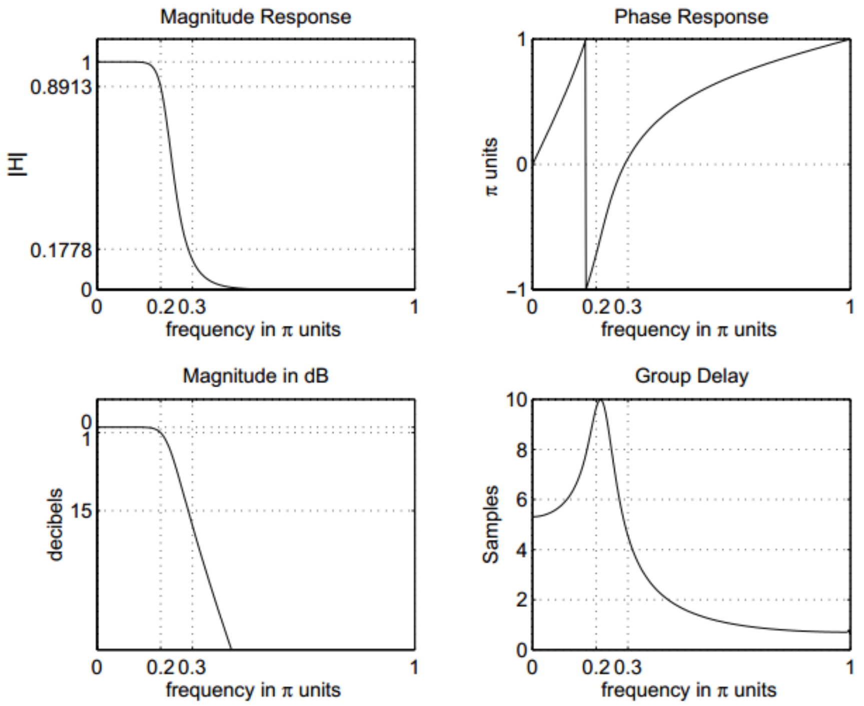


FIGURE 8.26 *Digital Butterworth lowpass filter using bilinear transformation*

EXAMPLE 8.18

Design the digital Chebyshev-I filter of Example 8.12. The specifications are

$$\omega_p = 0.2\pi,\ \ R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi,\ \ A_s = 15 \text{ dB}$$

Solutions

MATLAB script:

```
%% Example 8.18
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; % digital Passband freq in rad
```

```
ws = 0.3*pi; % digital Stopband freq in rad
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
% Analog Prototype Specifications: Inverse mapping for
frequencies
T = 1; Fs = 1/T; % Set T=1
OmegaP = (2/T)*tan(wp/2); % Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(ws/2); % Prewarp Prototype Stopband
freq
% Analog Chebyshev-1 Prototype Filter Calculation:
[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As);
% Bilinear transformation:
[b,a] = bilinear(cs,ds,Fs); [C,B,A] = dir2cas(b,a)

*** Chebyshev-1 Filter Order =  4
C =
   0.0018
B =
   1.0000    2.0000    1.0000
   1.0000    2.0000    1.0000
A =
   1.0000   -1.4996    0.8482
   1.0000   -1.5548    0.6493
```

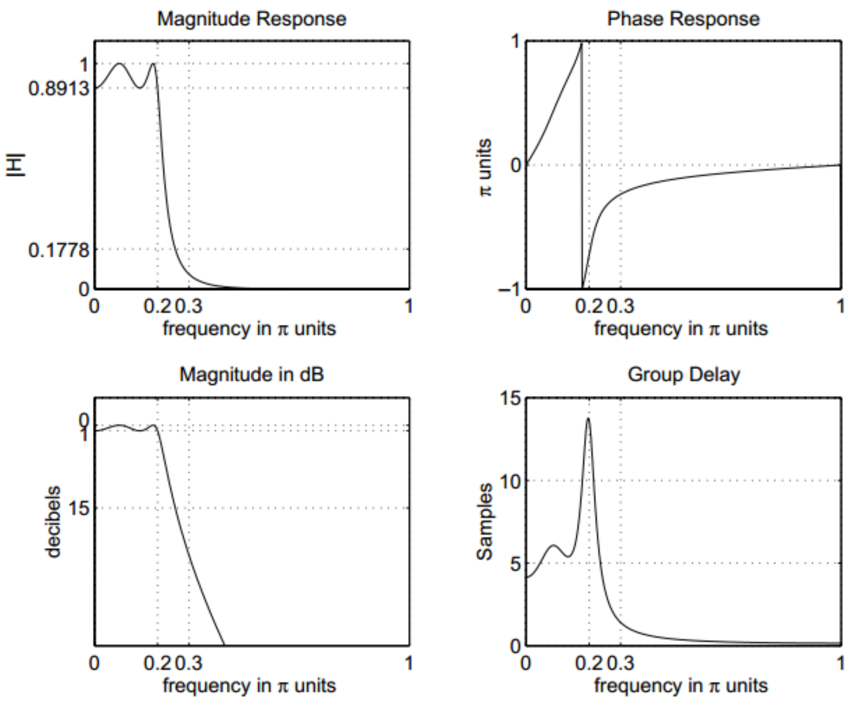


FIGURE 8.27 *Digital Chebyshev-I lowpass filter using bilinear transformation*

The desired filter is a 4th-order filter and has 4 zeros at $z = -1$. The system function is

$$H(z) = \frac{0.0018\left(1+z^{-1}\right)^4}{\left(1-1.4996z^{-1}+0.8482z^{-2}\right)\left(1-1.5548z^{-1}+0.6493z^{-2}\right)}$$

The frequency response plots are given in Figure 8.27 which are similar to those in Figure 8.22.

EXAMPLE 8.19

Design the digital Chebyshev-II filter of Example 8.13. The specifications are

$$\omega_p = 0.2\pi, \ R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi, \ A_s = 15 \text{ dB}$$

Solutions

MATLAB script:

```
%% Example 8.19
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; % digital Passband freq in rad
ws = 0.3*pi; % digital Stopband freq in rad
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
% Analog Prototype Specifications: Inverse mapping for
frequencies
T = 1; Fs = 1/T; % Set T=1
OmegaP = (2/T)*tan(wp/2); % Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(ws/2); % Prewarp Prototype Stopband
freq
% Analog Chebyshev-2 Prototype Filter Calculation:
[cs,ds] = afd_chb2(OmegaP,OmegaS,Rp,As);
% Bilinear transformation:
[b,a] = bilinear(cs,ds,Fs); [C,B,A] = dir2cas(b,a)

*** Chebyshev-2 Filter Order =  4
C =
    0.1797
B =

    1.0000    0.5574    1.0000
    1.0000   -1.0671    1.0000
A =
    1.0000   -0.4183    0.1503
    1.0000   -1.1325    0.7183
```

The desired filter is again a 4th-order filter with system function

$$H(z) = \frac{0.1797\left(1 + 0.5574z^{-1} + z^{-2}\right)\left(1 - 1.0671z^{-1} + z^{-2}\right)}{\left(1 - 0.4183z^{-1} + 0.1503z^{-2}\right)\left(1 - 1.1325z^{-1} + 0.7183z^{-2}\right)}$$

The frequency response plots are given in Figure 8.28. Note that the bilinear transformation has properly designed the Chebyshev-II digital filter.

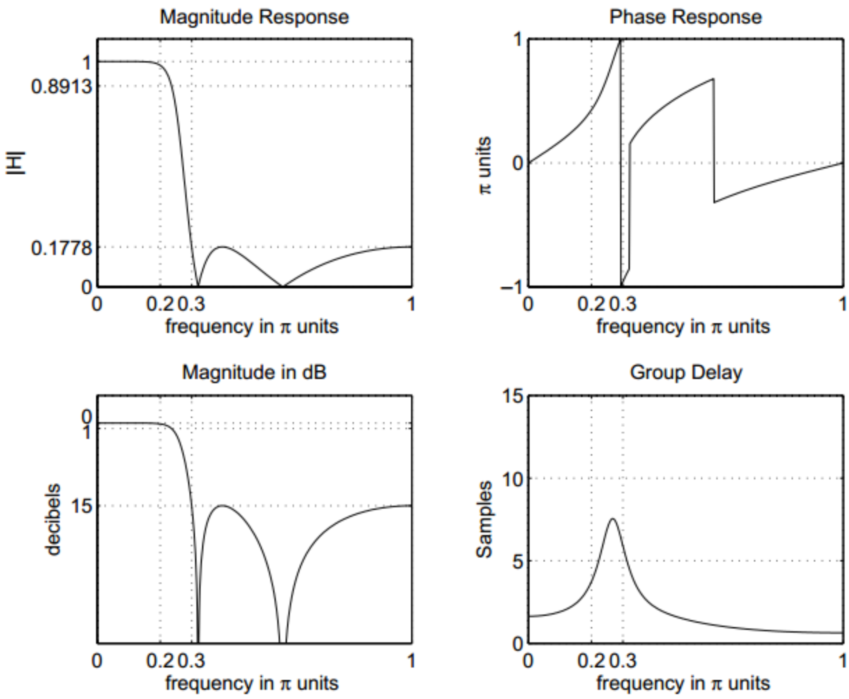


FIGURE 8.28 *Digital Chebyshev-II lowpass filter using bilinear transformation*

EXAMPLE 8.20

Design the digital elliptic filter of Example 8.14. The specifications are

$$\omega_p = 0.2\pi, \quad R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi, \quad A_s = 15 \text{ dB}$$

Solutions

MATLAB script:

```
%% Example 8.20
clear;clc;close all
% Digital Filter Specifications:
wp = 0.2*pi; % digital Passband freq in rad
ws = 0.3*pi; % digital Stopband freq in rad
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
% Analog Prototype Specifications: Inverse mapping for
```

```
frequencies
T = 1; Fs = 1/T; % Set T=1
OmegaP = (2/T)*tan(wp/2); % Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(ws/2); % Prewarp Prototype Stopband
freq
% Analog Elliptic Prototype Filter Calculation:
[cs,ds] = afd_elip(OmegaP,OmegaS,Rp,As);
% Bilinear transformation:
[b,a] = bilinear(cs,ds,Fs); [C,B,A] = dir2cas(b,a)

*** Elliptic Filter Order =  3
C =
    0.1214
B =
    1.0000   -1.4211    1.0000
    1.0000    1.0000         0
A =
    1.0000   -1.4928    0.8612
    1.0000   -0.6183         0
```

The desired filter is a 3rd-order filter with system function

$$H(z) = \frac{0.1214\left(1 - 1.4211z^{-1} + z^{-2}\right)\left(1 + z^{-1}\right)}{\left(1 - 1.4928z^{-1} + 0.8612z^{-2}\right)\left(1 - 0.6183z^{-1}\right)}$$

The frequency response plots are given in Figure 8.29. Note that the bilinear transformation has again properly designed the elliptic digital filter.

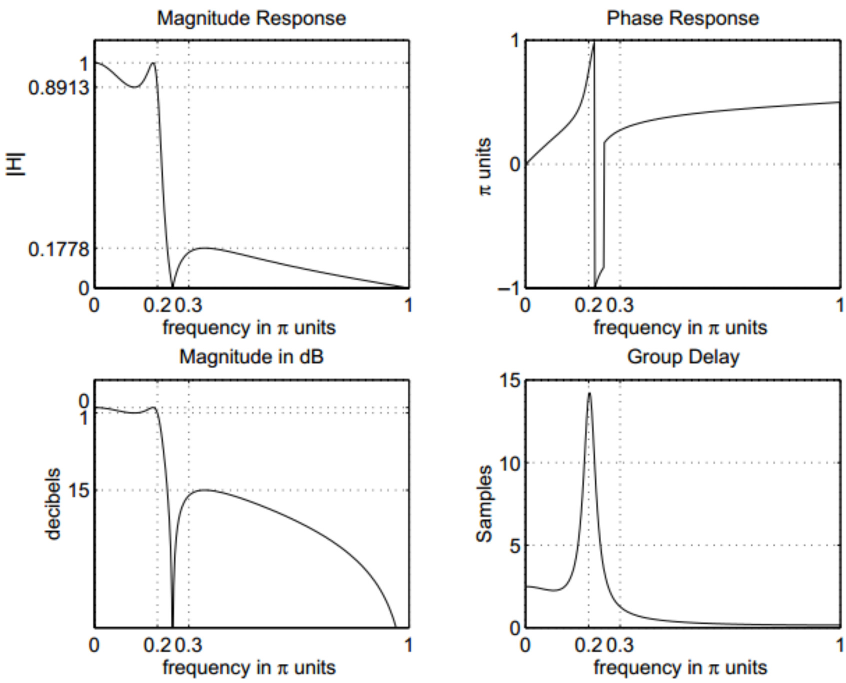


FIGURE 8.29 *Digital elliptic lowpass filter using bilinear transformation*

EXAMPLE 8.21

Digital Butterworth lowpass filter design:

$$\omega_p = 0.2\pi,\ R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi,\ A_s = 15 \text{ dB}$$

Solutions

```
%% Example 8.21
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; %digital Passband freq in rad
ws = 0.3*pi; %digital Stopband freq in rad
Rp = 1; %Passband ripple in dB
As = 15; %Stopband attenuation in dB
% Analog Prototype Specifications:
T = 1; %Set T=1
OmegaP = (2/T)*tan(wp/2); %Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(ws/2); %Prewarp Prototype Stopband
freq
% Analog Prototype Order Calculation:
N =ceil((log10((10^(Rp/10)-1)/(10^(As/10)-
1)))/(2*log10(OmegaP/OmegaS)));
fprintf('\n*** Butterworth Filter Order = %2.0f \n',N)
OmegaC = OmegaP/((10^(Rp/10)-1)^(1/(2*N))); %Analog BW
prototype cutoff
wn = 2*atan((OmegaC*T)/2); %Digital BW cutoff freq
% Digital Butterworth Filter Design:
wn = wn/pi; %Digital Butter cutoff in pi units
[b,a]=butter(N,wn); [b0,B,A] = dir2cas(b,a)

*** Butterworth Filter Order =  6
b0 =
   5.7969e-04
B =
    1.0000    2.0048    1.0048
    1.0000    1.9952    0.9952
    1.0000    2.0000    1.0000
A =
    1.0000   -0.9459    0.2342
    1.0000   -1.0541    0.3753
    1.0000   -1.3143    0.7149
```

The system function is

$$H(z) = \frac{0.00057969\left(1+z^{-1}\right)^6}{\left(1-0.9459z^{-1}+0.2342z^{-2}\right)\left(1-1.0541z^{-1}+0.3753z^{-2}\right)\left(1-1.3143z^{-1}+0.7149z^{-2}\right)}$$

which is the same as in Example 8.17. The frequency-domain plots were shown in Figure 8.26.

EXAMPLE 8.22

Digital Chebyshev-I lowpass filter design:

$$\omega_p = 0.2\pi,\ R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi,\ A_s = 15 \text{ dB}$$

Solutions

```
%% Example 8.22
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; %digital Passband freq in rad
ws = 0.3*pi; %digital Stopband freq in rad
Rp = 1; %Passband ripple in dB
As = 15; %Stopband attenuation in dB
% Analog Prototype Specifications:
T = 1; %Set T=1
OmegaP = (2/T)*tan(wp/2); %Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(ws/2); %Prewarp Prototype Stopband
freq
% Analog Prototype Order Calculation:
ep = sqrt(10^(Rp/10)-1); %Passband Ripple Factor
A = 10^(As/20); %Stopband Attenuation Factor
OmegaC = OmegaP; %Analog Prototype Cutoff freq
OmegaR = OmegaS/OmegaP; %Analog Prototype Transition
Ratio
g = sqrt(A*A-1)/ep; %Analog Prototype Intermediate cal.
N = ceil(log10(g+sqrt(g*g-
1))/log10(OmegaR+sqrt(OmegaR*OmegaR-1)));
fprintf('\n*** Chebyshev-1 Filter Order = %2.0f \n',N)
% Digital Chebyshev-I Filter Design:
wn = wp/pi; %Digital Passband freq in pi units
[b,a]=cheby1(N,Rp,wn); [b0,B,A] = dir2cas(b,a)

*** Chebyshev-1 Filter Order =  4
b0 =
```

```
    0.0018
B =
    1.0000    2.0000    1.0000
    1.0000    2.0000    1.0000
A =
    1.0000   -1.4996    0.8482
    1.0000   -1.5548    0.6493
```

The system function is

$$H(z) = \frac{0.0018\left(1+z^{-1}\right)^{4}}{\left(1-1.4996z^{-1}+0.8482z^{-2}\right)\left(1-1.5548z^{-1}+0.6493z^{-2}\right)}$$

which is the same as in Example 8.18. The frequency-domain plots were shown in Figure 8.27.

EXAMPLE 8.23

Digital Chebyshev-II lowpass filter design:

$$\omega_p = 0.2\pi,\ R_p = 1 \text{ dB}$$
$$\omega_s = 0.3\pi,\ A_s = 15 \text{ dB}$$

Solutions

```
%% Example 8.23
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; %digital Passband freq in rad
ws = 0.3*pi; %digital Stopband freq in rad
Rp = 1; %Passband ripple in dB
As = 15; %Stopband attenuation in dB
% Analog Prototype Specifications:
T = 1; %Set T=1
OmegaP = (2/T)*tan(wp/2); %Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(ws/2); %Prewarp Prototype Stopband
freq
% Analog Prototype Order Calculation:
ep = sqrt(10^(Rp/10)-1); %Passband Ripple Factor
A = 10^(As/20); %Stopband Attenuation Factor
OmegaC = OmegaP; %Analog Prototype Cutoff freq
OmegaR = OmegaS/OmegaP; %Analog Prototype Transition
Ratio
g = sqrt(A*A-1)/ep; %Analog Prototype Intermediate cal.
N = ceil(log10(g+sqrt(g*g-
```

```
1))/log10(OmegaR+sqrt(OmegaR*OmegaR-1)));
fprintf('\n*** Chebyshev-2 Filter Order = %2.0f \n',N)
% Digital Chebyshev-I Filter Design:
wn = ws/pi; %Digital Passband freq in pi units
[b,a]=cheby2(N,As,wn); [b0,B,A] = dir2cas(b,a)
```

```
*** Chebyshev-2 Filter Order =  4
b0 =
   0.1797
B =
   1.0000    0.5574    1.0000
   1.0000   -1.0671    1.0000
A =
   1.0000   -0.4183    0.1503
   1.0000   -1.1325    0.7183
```

The system function is

$$H(z) = \frac{0.1797\left(1+0.5574z^{-1}+z^{-2}\right)\left(1-1.0671z^{-1}+z^{-2}\right)}{\left(1-0.4183z^{-1}+0.1503z^{-2}\right)\left(1-1.1325z^{-1}+0.7183z^{-2}\right)}$$

which is the same as in Example 8.19. The frequency-domain plots were shown in Figure 8.28.

EXAMPLE 8.24

Digital elliptic lowpass filter design:

$$\omega_p = 0.2\pi,\ R_p = 1\text{ dB}$$
$$\omega_s = 0.3\pi,\ A_s = 15\text{ dB}$$

Solutions

```
%% Example 8.24
clear;clc;close all;
% Digital Filter Specifications:
wp = 0.2*pi; %digital Passband freq in rad
ws = 0.3*pi; %digital Stopband freq in rad
Rp = 1; %Passband ripple in dB
As = 15; %Stopband attenuation in dB
% Analog Prototype Specifications:
T = 1; %Set T=1
OmegaP = (2/T)*tan(wp/2); %Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(ws/2); %Prewarp Prototype Stopband
freq
% Analog Elliptic Filter order calculations:
```

```
ep = sqrt(10^(Rp/10)-1); %Passband Ripple Factor
A = 10^(As/20); %Stopband Attenuation Factor
OmegaC = OmegaP; %Analog Prototype Cutoff freq
k = OmegaP/OmegaS; %Analog Prototype Transition Ratio;
k1 = ep/sqrt(A*A-1); %Analog Prototype Intermediate cal.
capk = ellipke([k.^2 1-k.^2]);
capk1 = ellipke([(k1 .^2) 1-(k1 .^2)]);
N = ceil(capk(1)*capk1(2)/(capk(2)*capk1(1)));
fprintf('\n*** Elliptic Filter Order = %2.0f \n',N)
% Digital Elliptic Filter Design:
wn = wp/pi; %Digital Passband freq in pi units
[b,a]=ellip(N,Rp,As,wn); [b0,B,A] = dir2cas(b,a)
```

```
*** Elliptic Filter Order =  3
b0 =
    0.1214
B =
    1.0000   -1.4211    1.0000
    1.0000    1.0000         0
A =
    1.0000   -1.4928    0.8612
    1.0000   -0.6183         0
```

The system function is

$$H(z) = \frac{0.1214\left(1 - 1.4211z^{-1} + z^{-2}\right)\left(1 + z^{-1}\right)}{\left(1 - 1.4928z^{-1} + 0.8612z^{-2}\right)\left(1 - 0.6183z^{-1}\right)}$$

which is the same as in Example 8.20. The frequency-domain plots were shown in Figure 8.29.

TABLE 8.1 *Comparison of three filters*

Prototype	*Order N*	*Stopband Att.*
Butterworth	6	15
Chebyshev-I	4	25
Elliptic	3	27

EXAMPLE 8.25

In Example 8.22 we designed a Chebyshev-I lowpass filter with specifications

$$\omega'_p = 0.2\pi, \quad R_p = 1 \text{ dB}$$
$$\omega'_s = 0.3\pi, \quad A_s = 15 \text{ dB}$$

and determined its system function

$$H_{LP}(Z) = \frac{0.001836(1+Z^{-1})^4}{(1-1.4996Z^{-1}+0.8482Z^{-2})(1-1.5548Z^{-1}+0.6493Z^{-2})}$$

Design a highpass filter with these tolerances but with passband beginning at $\omega_p = 0.6\pi$.

Solutions

We want to transform the given lowpass filter into a highpass filter such that the cutoff frequency $\omega'_p = 0.2\pi$ is mapped onto the cutoff frequency $\omega_p = 0.6\pi$.
From Table 8.2

$$\alpha = -\frac{\cos[(0.2\pi + 0.6\pi)/2]}{\cos[(0.2\pi - 0.6\pi)/2]} = -0.38197 \qquad \textbf{(8.70)}$$

Hence

$$\begin{aligned} H_{LP}(z) &= H(Z)|_{Z=-\frac{z^{-1}-0.38197}{1-0.38197z^{-1}}} \\ &= \frac{0.02426(1-z^{-1})^4}{(1+0.5661z^{-1}+0.7657z^{-2})(1+1.0416z^{-1}+0.4019z^{-2})} \end{aligned}$$

which is the desired filter. The frequency response plots of the lowpass filter and the new highpass filter are shown in Figure 8.31.

TABLE 8.2 *Frequency transformation for digital filters (prototype lowpass filter has cutoff frequency ω'_c)*

Type of Transformation	*Transformation*	*Parameters*
Lowpass	$z^{-1} \longrightarrow \frac{z^{-1}-\alpha}{1-\alpha z^{-1}}$	ω_c = cutoff frequency of new filter $\alpha = \frac{\sin[(\omega'_c - \omega_c)/2]}{\sin[(\omega'_c + \omega_c)/2]}$
Highpass	$z^{-1} \longrightarrow -\frac{z^{-1}+\alpha}{1+\alpha z^{-1}}$	ω_c = cutoff frequency of new filter $\alpha = -\frac{\cos[(\omega'_c + \omega_c)/2]}{\cos[(\omega'_c - \omega_c)/2]}$
Bandpass	$z^{-1} \longrightarrow -\frac{z^{-2}-\alpha_1 z^{-1}+\alpha_2}{\alpha_2 z^{-2}-\alpha_1 z^{-1}+1}$	ω_ℓ = lower cutoff frequency ω_u = upper cutoff frequency $\alpha_1 = -2\beta K/(K+1)$ $\alpha_2 = (K-1)/(K+1)$ $\beta = \frac{\cos[(\omega_u + \omega_\ell)/2]}{\cos[(\omega_u - \omega_\ell)/2]}$ $K = \cot\frac{\omega_u - \omega_\ell}{2}\tan\frac{\omega'_c}{2}$
Bandstop	$z^{-1} \longrightarrow \frac{z^{-2}-\alpha_1 z^{-1}+\alpha_2}{\alpha_2 z^{-2}-\alpha_1 z^{-1}+1}$	ω_ℓ = lower cutoff frequency ω_u = upper cutoff frequency $\alpha_1 = -2\beta/(K+1)$ $\alpha_2 = (K-1)/(K+1)$ $\beta = \frac{\cos[(\omega_u + \omega_\ell)/2]}{\cos[(\omega_u - \omega_\ell)/2]}$ $K = \tan\frac{\omega_u - \omega_\ell}{2}\tan\frac{\omega'_c}{2}$

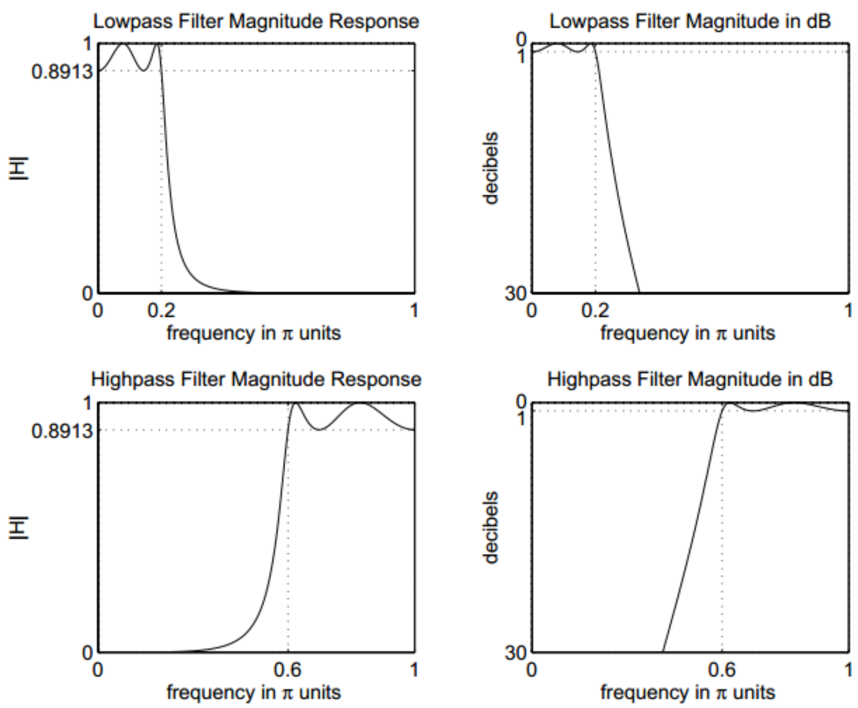


FIGURE 8.31 *Magnitude response plots for Example 8.25*

EXAMPLE 8.26

Use the **zmapping** function to perform the lowpass-to-highpass transformation in Example 8.25.

Solutions

First we will design the lowpass digital filter in MATLAB using the bilinear transformation procedure and then use the **zmapping** function. MATLAB script:

```
%% Example 8.26
clear;clc;close all;
% Digital Lowpass Filter Specifications:
wplp = 0.2*pi; % digital Passband freq in rad
wslp = 0.3*pi; % digital Stopband freq in rad
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
% Analog Prototype Specifications: Inverse mapping for
frequencies
T = 1; Fs = 1/T; % Set T=1
OmegaP = (2/T)*tan(wplp/2); % Prewarp Prototype Passband
freq
OmegaS = (2/T)*tan(wslp/2); % Prewarp Prototype Stopband
freq
```

```
% Analog Chebyshev Prototype Filter Calculation:
[cs,ds] = afd_chb1(OmegaP,OmegaS,Rp,As);
% Bilinear transformation:
[blp,alp] = bilinear(cs,ds,Fs);
% Digital Highpass Filter Cutoff frequency:
wphp = 0.6*pi; % Passband edge frequency
% LP-to-HP frequency-band transformation:
alpha = -(cos((wplp+wphp)/2))/(cos((wplp-wphp)/2))
Nz = -[alpha,1]; Dz = [1,alpha];
[bhp,ahp] = zmapping(blp,alp,Nz,Dz); [C,B,A] =
dir2cas(bhp,ahp)

*** Chebyshev-1 Filter Order =  4
alpha =
  -0.3820
C =
    0.0243
B =
    1.0000   -2.0000    1.0000
    1.0000   -2.0000    1.0000
A =
    1.0000    1.0416    0.4019
    1.0000    0.5561    0.7647
```

The system function of the highpass filter is

$$H(z) = \frac{0.0243(1-z^{-1})^4}{(1+0.5661z^{-1}+0.7647z^{-2})(1+1.0416z^{-1}+0.4019z^{-2})}$$

which is essentially identical to that in Example 8.25.

EXAMPLE 8.27

Design a highpass digital filter to satisfy

$$\omega_p = 0.6\pi,\ R_p = 1 \text{ dB}$$
$$\omega_s = 0.4586\pi,\ A_s = 15 \text{ dB}$$

Use the Chebyshev-I prototype.

Solutions

MATLAB script:

```
%% Example 8.27
clear;clc;close all;
% Digital Highpass Filter Specifications:
wp = 0.6*pi; % digital Passband freq in rad
```

```
ws = 0.4586*pi; % digital Stopband freq in rad
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
[b,a] = cheb1hpf(wp,ws,Rp,As); [C,B,A] = dir2cas(b,a)

*** Chebyshev-1 Filter Order =  4
C =
    0.0243
B =
    1.0000   -2.0000    1.0000
    1.0000   -2.0000    1.0000
A =
    1.0000    1.0416    0.4019
    1.0000    0.5561    0.7647
```

The system function is

$$H(z) = \frac{0.0243(1 - z^{-1})^4}{(1 + 0.5661z^{-1} + 0.7647z^{-2})(1 + 1.0416z^{-1} + 0.4019z^{-2})}$$

which is identical to that in Example 8.26.

EXAMPLE 8.28

In this example we will design a Chebyshev-I highpass filter whose specifications were given in Example 8.27.

Solutions

MATLAB script:

```
%% Example 8.28
clear;clc;close all;
% Digital Filter Specifications: % Type: Chebyshev-I
highpass
ws = 0.4586*pi; % Dig. stopband edge frequency
wp = 0.6*pi; % Dig. passband edge frequency
Rp = 1; % Passband ripple in dB
As = 15; % Stopband attenuation in dB
% Calculations of Chebyshev-I Filter Parameters:
[N,wn] = cheb1ord(wp/pi,ws/pi,Rp,As);
% Digital Chebyshev-I Highpass Filter Design:
[b,a] = cheby1(N,Rp,wn,'high');
% Cascade Form Realization:
[b0,B,A] = dir2cas(b,a)
```

```
b0 =
    0.0243
B =
    1.0000   -2.0000    1.0000
    1.0000   -2.0000    1.0000
A =
    1.0000    1.0416    0.4019
    1.0000    0.5561    0.7647
```

The cascade form system function

$$H(z) = \frac{0.0243(1 - z^{-1})^4}{(1 + 0.5661z^{-1} + 0.7647z^{-2})(1 + 1.0416z^{-1} + 0.4019z^{-2})}$$

is identical to the filter designed in Example 8.27, which demonstrates that the two approaches described on page 386 are identical. The frequency-domain plots are shown in Figure 8.32.

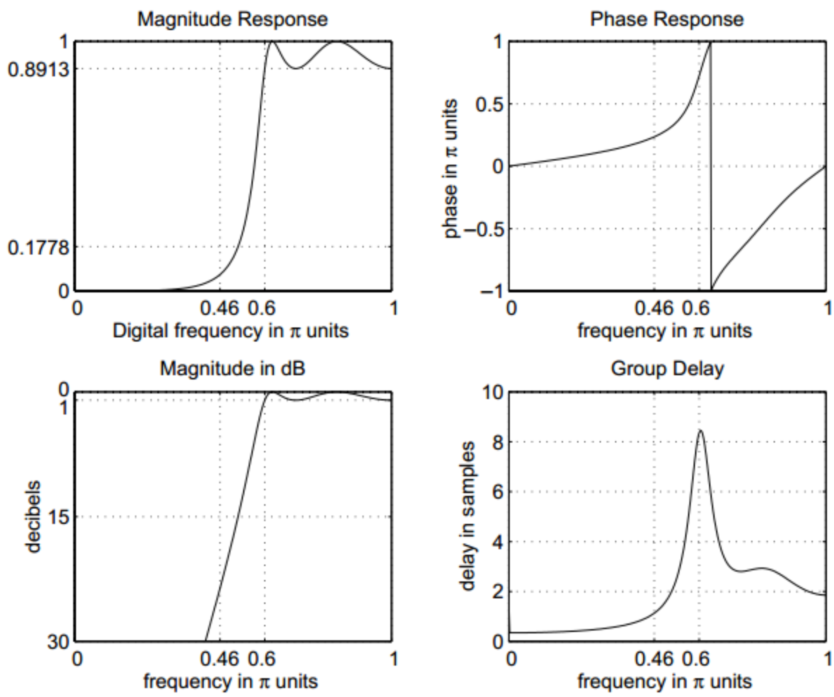


FIGURE 8.32 *Digital Chebyshev-I highpass filter in Example 8.28*

EXAMPLE 8.29

In this example we will design an elliptic bandpass filter whose specifications are given in the following MATLAB script:

Solutions

```
%% Example 8.29
```

```
clear;clc;close all;
% Digital Filter Specifications: % Type: Elliptic
Bandpass
ws = [0.3*pi 0.75*pi]; % Dig. stopband edge frequency
wp = [0.4*pi 0.6*pi]; % Dig. passband edge frequency
Rp = 1; % Passband ripple in dB
As = 40; % Stopband attenuation in dB
% Calculations of Elliptic Filter Parameters:
[N,wn] = ellipord(wp/pi,ws/pi,Rp,As);
% Digital Elliptic Bandpass Filter Design:
[b,a] = ellip(N,Rp,As,wn);
% Cascade Form Realization:
[b0,B,A] = dir2cas(b,a)

b0 =
   0.0197
B =
   1.0000    1.5066    1.0000
   1.0000    0.9269    1.0000
   1.0000   -0.9269    1.0000
   1.0000   -1.5066    1.0000
A =
   1.0000    0.5963    0.9399
   1.0000    0.2774    0.7929
   1.0000   -0.2774    0.7929
   1.0000   -0.5963    0.9399
```

Note that the designed filter is a 10th-order filter. The frequency-domain plots are shown in Figure 8.33.

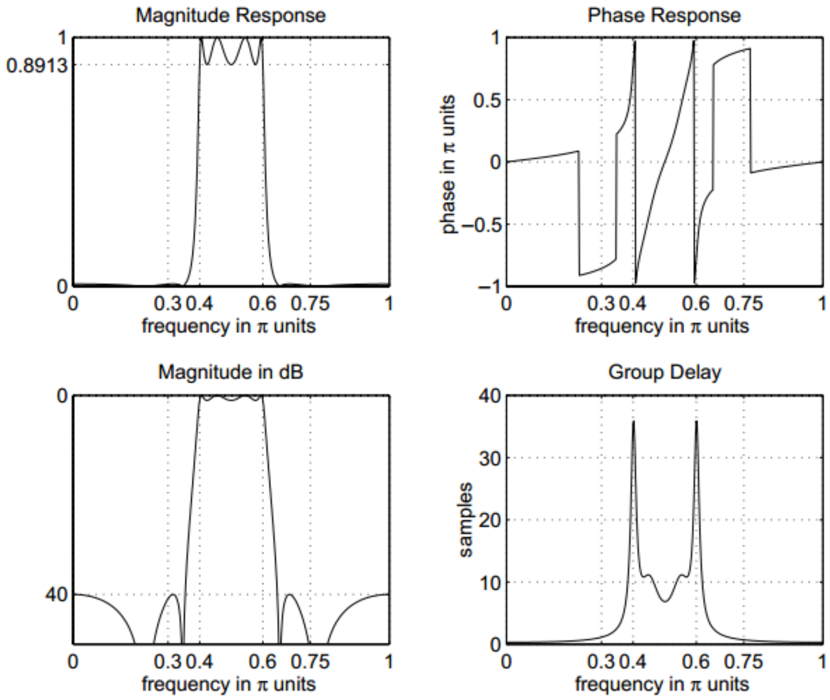


FIGURE 8.33 *Digital elliptic bandpass filter in Example 8.29*

EXAMPLE 8.30

Finally, we will design a Chebyshev-II bandstop filter whose specifications are given in the following MATLAB script.

Solutions

```
%% Example 8.30
clear;clc;close all;
% Digital Filter Specifications: % Type: Chebyshev-II
Bandstop
ws = [0.4*pi 0.7*pi]; % Dig. stopband edge frequency
wp = [0.25*pi 0.8*pi]; % Dig. passband edge frequency
Rp = 1; % Passband ripple in dB
As = 40; % Stopband attenuation in dB
% Calculations of Chebyshev-II Filter Parameters:
[N,wn] = cheb2ord(wp/pi,ws/pi,Rp,As);
% Digital Chebyshev-II Bandstop Filter Design:
[b,a] = cheby2(N,As,ws/pi,'stop');
% Cascade Form Realization:
[b0,B,A] = dir2cas(b,a)

b0 =
```

```
    0.1558
B =
    1.0000    1.1456    1.0000
    1.0000    0.8879    1.0000
    1.0000    0.3511    1.0000
    1.0000   -0.2434    1.0000
    1.0000   -0.5768    1.0000
A =
    1.0000    1.3041    0.8031
    1.0000    0.8901    0.4614
    1.0000    0.2132    0.2145
    1.0000   -0.4713    0.3916
    1.0000   -0.8936    0.7602
```

This is also a 10th-order filter. The frequency domain plots are shown in Figure 8.34.

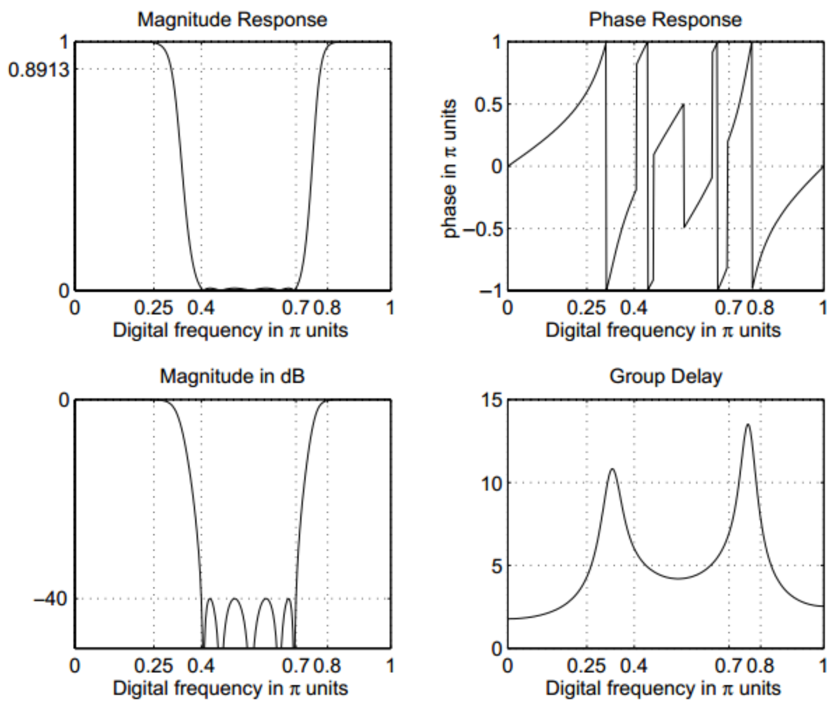


FIGURE 8.34 *Digital Chebyshev-II bandstop filter in Example 8.30*